NASA CR-159,123

3 1176 00138 6292

NASA CONTRACTOR REPORT 159123

NASA-CR-159123

1980 0007164

CARE III FINAL REPORT

PHASE I

VOLUME II

J. J. Stiffler, L. A. Bryant, L. Guccione

RAYTHEON COMPANY
SUDBURY, MASSACHUSETTS 01776

PREPARED UNDER
NASA CONTRACT NAS1-15072

FOR
NASA LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA

AIR FORCE AVIONICS LABORATORY
WRIGHT PATTERSON AIR FORCE BASE, OHIO

NOVEMBER 1979

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665
AC 804 827-3966

NASA CONTRACTOR REPORT 159123

CARE III FINAL REPORT

PHASE I

VOLUME II

J. J. Stiffler, L. A. Bryant, L. Guccione

RAYTHEON COMPANY
SUDBURY, MASSACHUSETTS 01776

PREPARED UNDER
NASA CONTRACT NAS1-15072

FOR
NASA LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA

AIR FORCE AVIONICS LABORATORY
WRIGHT PATTERSON AIR FORCE BASE, OHIO

NOVEMBER 1979

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665
AC 804 827-3966

N80-15424 #

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-159122		
4. Title and Subtitle CARE III Final Report, Phase I, Volume II		5. Report Date November 1979 6. Performing Organization Code
7. Author(s) J. J. Stiffler, L. A. Bryant, L. Guccione		8. Performing Organization Report No. 10. Work Unit No.
9. Performing Organization Name and Address Raytheon Company Sudbury, MA 01776		11. Contract or Grant No. NAS1-15072 13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center, Hampton, VA 23665 Wright Patterson Air Force Base Air Force Avionics Laboratory, Dayton, Ohio		14. Sponsoring Agency Code LARC, WPAFB
15. Supplementary Notes NASA, Project Engineer, Salvatore J. Bavuso WPAFB, Technical Monitor, Lt. Barry Baxley		

16. Abstract

This report describes the work done during the first phase of a two-phase effort to develop a computer program to aid in assessing the reliability of fault-tolerant avionics systems. The overall effort consists of five major tasks: 1) Establish the basic requirements that must be satisfied if the program is to achieve its overall objective. 2) Define a general program structure consistent with these requirements. 3) Develop and program a mathematical model relating the reliability of a fault-tolerant system to the (not necessarily time-independent) failure rates and coverage factors characterizing its various elements. 4) Develop and program a mathematical model for evaluating the coverage (probability of successful recovery) associated with any given fault as a function of the type and location of the fault, the applicable fault detection and isolation mechanism, and the number and status of prior faults. 5) Develop and program a procedure whereby a user of these models can accurately and conveniently specify the configuration of the system to be evaluated and the constraints influencing its ability to recover from faults.

The first three of these tasks were completed during Phase One; the resulting requirements, program structure, and reliability model are discussed in detail in Volume I of this report, along with the tradeoffs and sample reliability assessments made in arriving at the approach finally taken. The Computer Program Requirements Document is contained in Volume II. This latter volume also includes several appendices containing computer print-outs and other ancillary material supporting the conclusions presented in Volume I.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Fault-Tolerant Avionics Systems, Reliability Modeling, Fault Coverage, Fault Models	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified		

* For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

CARE III COMPUTER PROGRAM REQUIREMENTS DOCUMENT

1.0 SCOPE

1.1 IDENTIFICATION

This specification establishes requirements for performance, design, test and evaluation of the Computer-Aided Reliability Estimation System (CARE III).

1.2 FUNCTIONAL SUMMARY

The CARE III system will be a general-purpose fault-tolerant computer reliability estimation tool. It will be written based on a Modularized Direct Access Information System containing three main modules:

a. Batch or interactive input processor: CAREINB or CAREINI

b. Coverage calculator: COVRGE

c. Reliability estimator: CARE3

10-1349 (11-68) PRINTED IN U. S. A.

2.0 APPLICABLE DOCUMENTS

1. CARE III Final Report, Phase 1; ER79-4102; 18 April 1979

2. CARE II Reliability Model Final Report, ER74-4108, 25 March 1974

10-1349 (11/68) PRINTED IN U. S. A.

3.0 REQUIREMENTS

3.1 SYSTEM PERFORMANCE

The CARE III system will provide an interactive or batch environment for data input. It will also provide the capability of modeling at least 40 stages with N state-operating modes and multiple dependencies across stages, i.e., n-coupled stages.

To perform as specified, a modular design for the system is being proposed. Due to the recursive processing required when computing the computer configuration's reliability, certain storage limitations occur as the number of coupled stages increases. These items are detailed in the following subsections, 3.1.1 and 3.1.2.

3.1.1 MODULAR DIRECT ACCESS INFORMATION SYSTEM DESIGN

The CARE III system can be written and executed very efficiently if the system is split into independent modules. In this manner, core requirements are kept at a minimum because at any one point the only section of the system loaded into core for execution is the section currently required. Splitting the system into separate programs representing specific functions also readily organizes for the user exactly what set of inputs is currently required. Then, given a modularized design, the input processor routine can interactively generate files for subsequent reliability runs in the batch mode. These files can be made permanent disk files for later modification runs.

With this approach, it is recommended that the CARE III system consist of three main programs: CAREIN($\frac{I}{B}$), COVRGE and CARE3.

10-1349 (11-68) PRINTED IN U. S. A.

The main function of routine CAREIN will be to preprocess all inputs required for defining the type of computer configuration to be modeled, and to preprocess the necessary inputs required to define coverage functions and Detection/Isolation/Recovery mechanisms. This system definition input will be stored in a random (word-addressable) file through the use of mass storage input/output (MSIO) subroutines controlled by CDC Record Manager. This file will later be accessed by programs COVRGE and CARE3 and used to compute the model's reliability. This file will also contain all non-overridden parameter defaults and can be later modified without having to completely redefine the model.

The following describes one such usage of this "Direct Access Information System" approach for a series of four computer runs using a batch environment.

RUN #1 - Define the computer configuration and have its reliability computed.

RUN #2 - Add coverage values to the model and have the reliability computed.

RUN #3 - Define Detection/Isolation/Recovery functions, and test the model's reliability by defining certain D/I/R mechanisms using these functions.

RUN #4 - Define a second set of D/I/R mechanisms using the previously defined functions, and test the model's reliability.

If the input program was run in an interactive environment, the model would first be defined interactively and then the reliability computation program could be submitted for execution in the batch mode. Later modifications or additions to the model could then be made interactively, and the program submitted for execution again.

10-1349 (11 68) PRINTED IN U. S. A.

This modularized approach will cut operational costs due to the random file input approach, which will save data between runs, thus eliminating the need to reinput data. Also core size will be reduced by this approach which leads to a cost savings.

3.1.2 STORAGE LIMITATIONS

The CARE3 program recursively computes each subsystem's reliability based on the computation of each state vector's probability; where the subsystem's state vectors represent combinations of possible failed units. The loop structure required to compute all required state vectors is based upon computing "sets" of state vectors (see CARE III Final Report, Section 4.4.2). This structure requires that only two sets of state vectors be in memory at any one time. Even with this complex method for defining the recursive state vectors, the array (denoted QLT) required to contain these two sets becomes enormous as the number of coupled-stages per subsystem increases.

The capability of modeling up to 40 stages can be met by concatenating several runs, each run modeling fewer than 40 stages. In order to couple n-stages, the maximum of the number of failed units allowable in each stage (denoted MFU) + 1 is used to determine the maximum possible coupled-stages. As an upper bound on the QLT array size, $100{,}000_8$ (32768_{10}) words will be used in the following equation to determine maximum n-stages (coupled) given MFU for 51 time steps per vector.

The upper bound equation is $MFU^n - (MFU-2)^n \leq 100{,}000_8/51$ time steps per vector which yields the following chart for MFU versus n-stages (coupled).

MFU per Stage	n-stages (Coupled)
2	9
3	5
4	4
5	4
6	3
7	3
8	3
9	3
10	3
11	3
12	2
⋮	⋮
MAX (TBSL)	2

Chart 3-1

The above equation assumes that each stage per subsystem can have MFU failures. A utility routine MAXCPLD exists to determine if the QLT array will overflow for a given number of stages having a specified number of units per stage, survivors per stage and number of time steps.

For illustrative purposes, the following chart shows a computer system with eight subsystems; i.e., eight concatenated runs are required to compute this system's reliability and the number of coupled stages allowed given the required MFU's.

3.2 ENVIRONMENT

3.2.1 EQUIPMENT CONFIGURATION

The CARE III system shall be written to run on CDC computers which support CDC FORTRAN Extended 4 language. The disk files used within the system will be random, word addressable files

10-1349 (11/68) PRINTED IN U. S. A.

Stage	Initial Configuration (No. of Units)		No. of Survivor Units	No. of Possible Failed Units (including 0 failures)	Maximum of the Failed Units (MFU) per stage
1	15	*	2	14	
2	9	coupled subsystem 1	2	8	14
3	5		2	4	
4	7		3	5	
5	5	coupled subsystem 2	2	4	5
6	5		1	5	
7	4		2	3	
8	2	success path #1	1	2	
9	3		2	2	
10	4	coupled subsystem 3	2	3	3
11	3		1	3	
12	3	success path #2	1	3	
13	8		3	6	
14	7	coupled subsystem 4	2	6	6
15	6		2	5	
16	20	coupled subsystem 5	5	16	16
17	15		3	13	
18	10	independent subsystem 6	5	6	6
19	8	independent subsystem 7	3	6	6
20	5	independent subsystem 8	2	4	4

*NOTE: Subsystem 1 has three coupled-stages even though Chart 3-1 shows that for an MFU of 14 only two stages can be coupled. Because the other stages in this subsystem allow a lot fewer than 14 possible failed units, it is possible to couple these three stages.

Chart 3-2 EXAMPLE OF COUPLING CAPABILITY

controlled by CDC Record Manager.

3.2.2 SOFTWARE CONFIGURATION

The CARE III system shall consist of four main FORTRAN programs, two of which will be the interactive and batch versions of the input processor; the third one will be the coverage model; the fourth one will compute the reliability of the specified computer system being modeled. Each main program shall have a complement of subroutines and functions written mainly in FORTRAN Extended 4 language. A minimal number of routines shall be written in CDC COMPASS 3 language.

3.2.3 INTERFACES

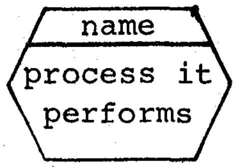
The following flow diagrams depict the proposed design of the CARE III system.

Two text input files are required: one to define the computer configuration and one to aid in the calculation of the coverage model. If coverage is preset per state in the configuration file INFILE, the coverage input file CVFILE need not be defined by the user.

The Direct Access Information System (DAIS) files generated by CARE III are designed to be random, word addressable mass storage files. Each record within these files can be accessed with a master index or subindex(es). The DAIS files will contain the processed user input required by programs COVRGE and CARE3. They will be made permanent disk files by CARE III so that they can be modified if desired without having to reinput the entire data set. Thus a second run can use existing files CAREDF and CARECV, after minor modifications have been made to them, by running program CARIN using only an updated portion of the original inputs. This capability is especially

10-1349 (11 68) PRINTED IN U. S. A.

convenient if the user runs the interactive CAREIN program.

The symbol 

denotes a separate routine for which a separate flow diagram exists in the pages following.

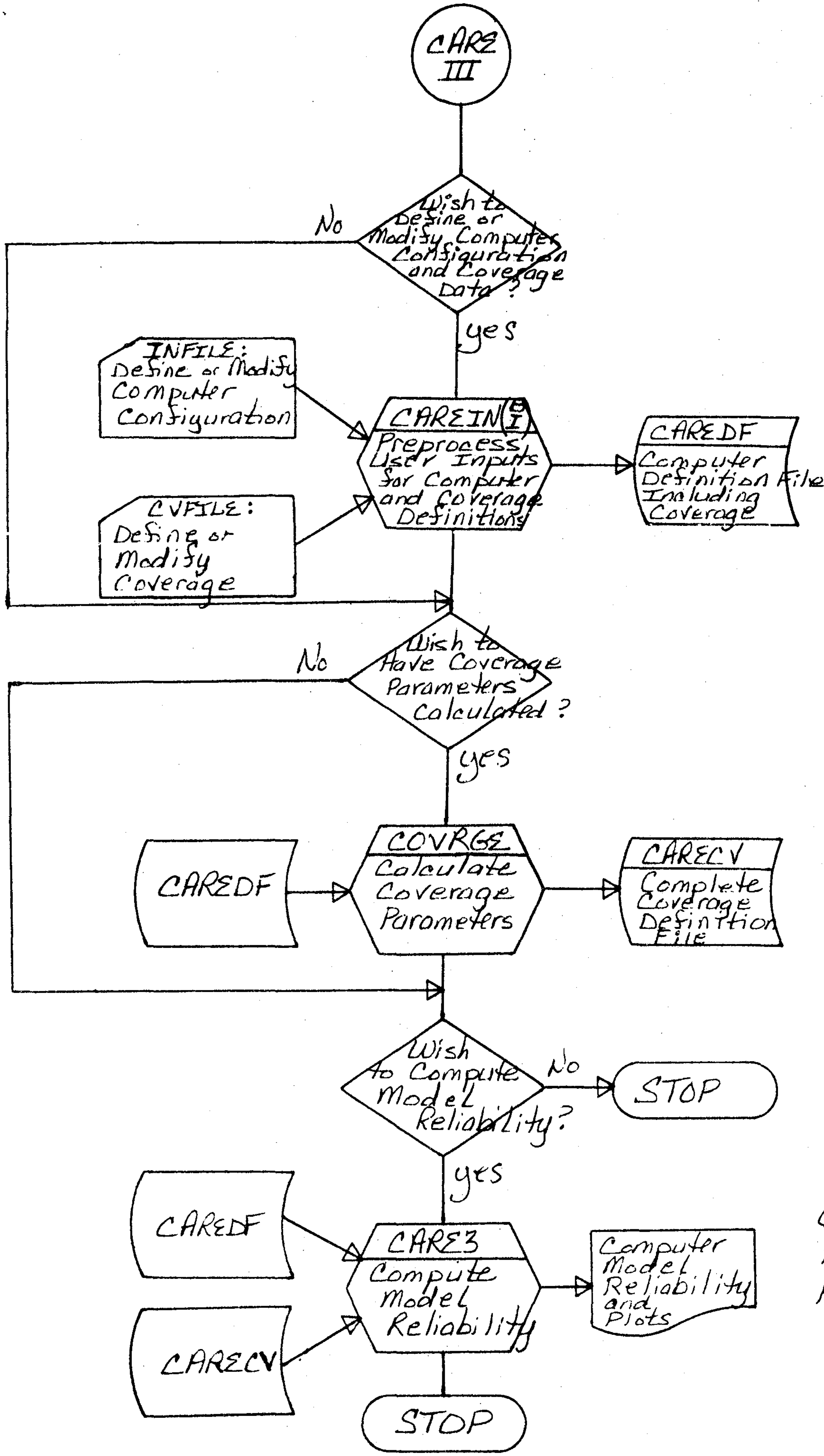
The CAREIN flow diagrams describe the processing of the NAMELIST input commands required, and the creation of the DAIS files. The flow diagrams for COVRGE and CAREIN detail the processing of the DAIS files.

3.3 FUNCTIONAL REQUIREMENTS

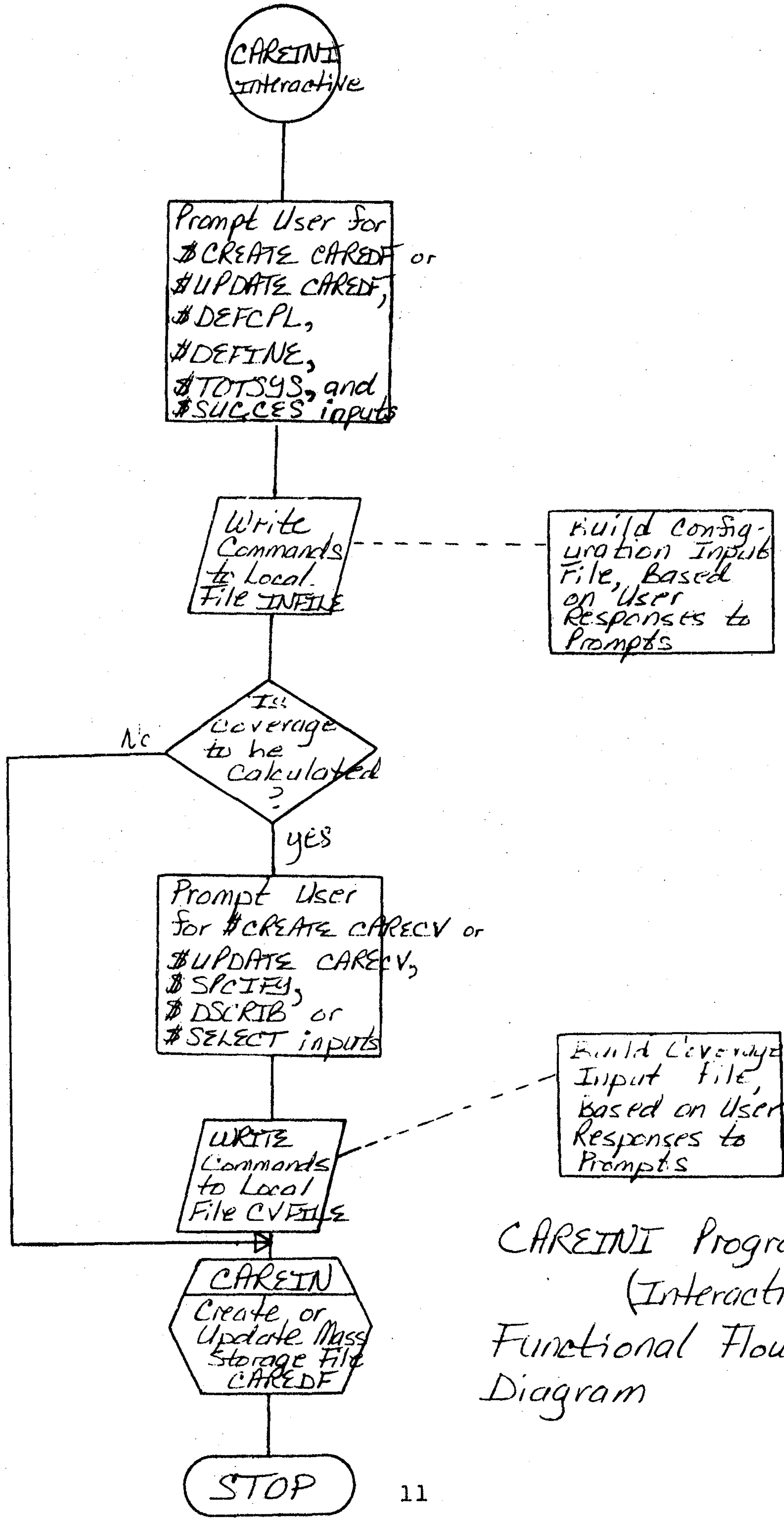
3.3.1 CAREIN INPUT PROCESSOR USING NAMELISTS FOR INPUT COMMANDS

Commands to the CARE system input processor routine CAREIN (Interactive or Batch) will be in the form of FORTRAN NAMELIST groups. The NAMELIST feature of FORTRAN will be used as an input template rather than as a way of defining actual variables in the program. Each necessary input command to CAREIN will be in the form of its corresponding NAMELIST definition within the program. Because each NAMELIST can be used over and over to define the necessary inputs, the user has a general form in which to define all of the data. The program then transfers the data to the CAREDF random access file and the NAMELIST variables are cleared for the next input card.

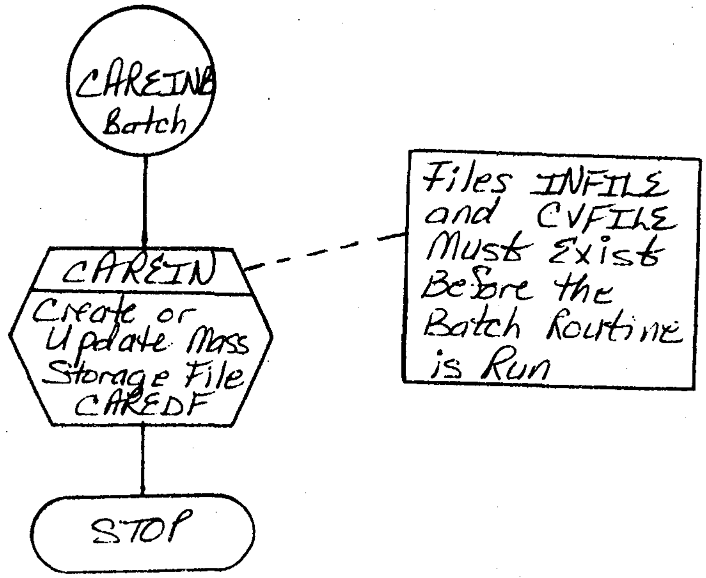
The NAMELIST groups are set up so that each one specifies a different set of commands necessary to run the program.



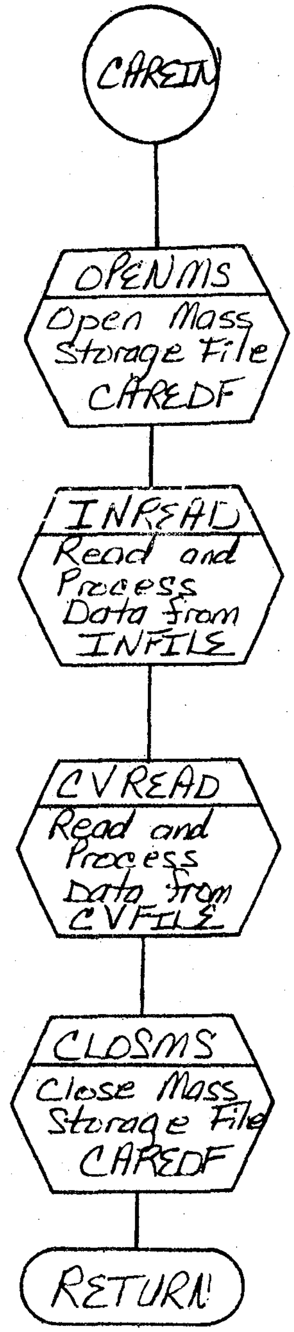
CARE III System Functional Flow-Diagram



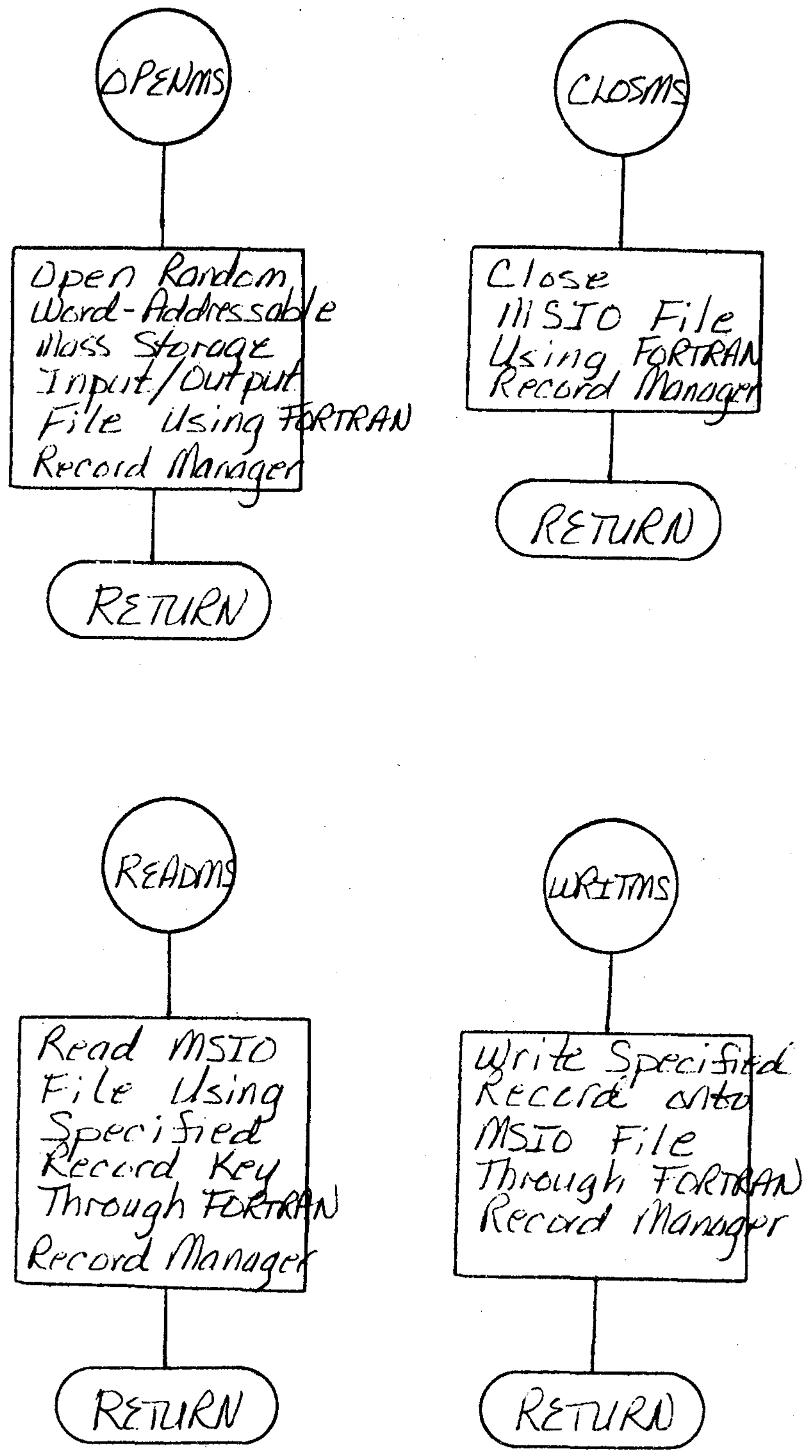
CAREINI Program (Interactive) Functional Flow-Diagram



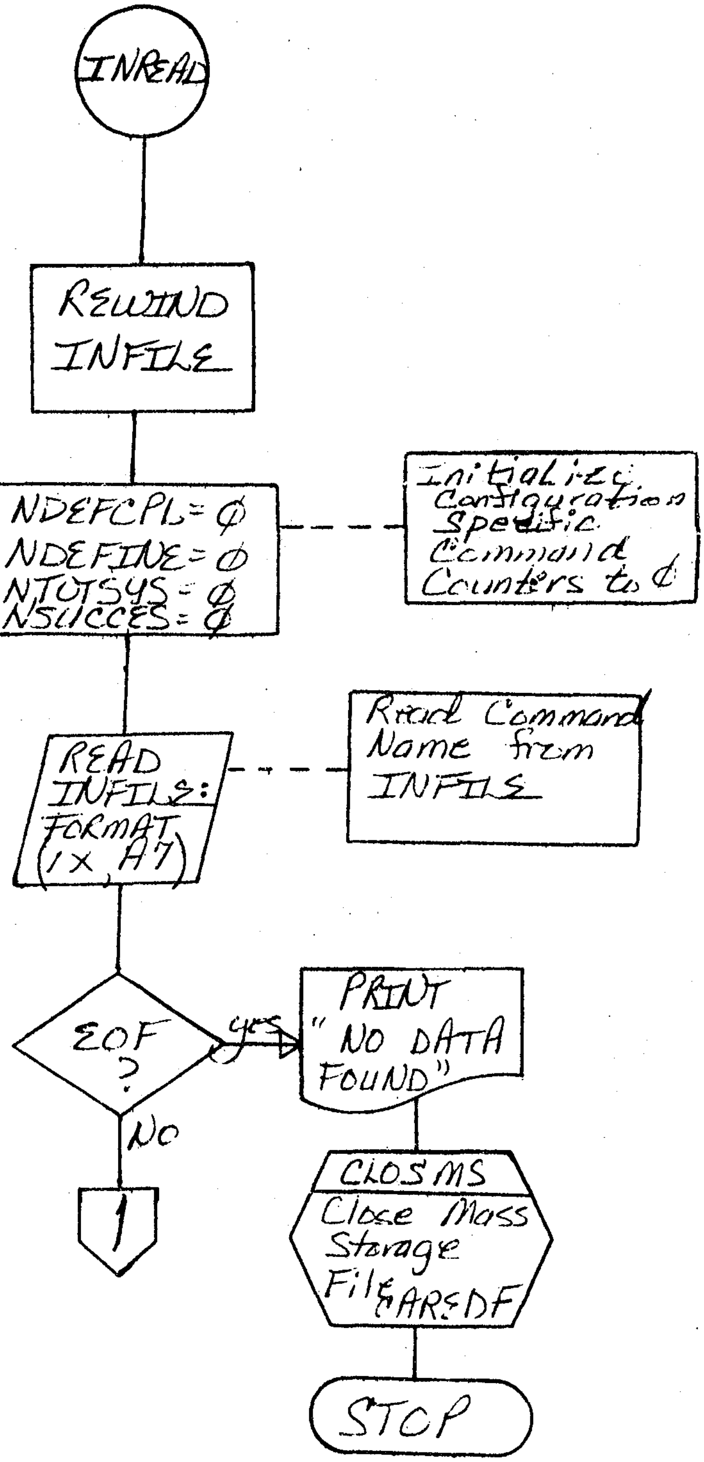
CAREINB Program (Batch) Functional Flow-Diagram



CAREIN Subroutine Functional Flow-Diagram

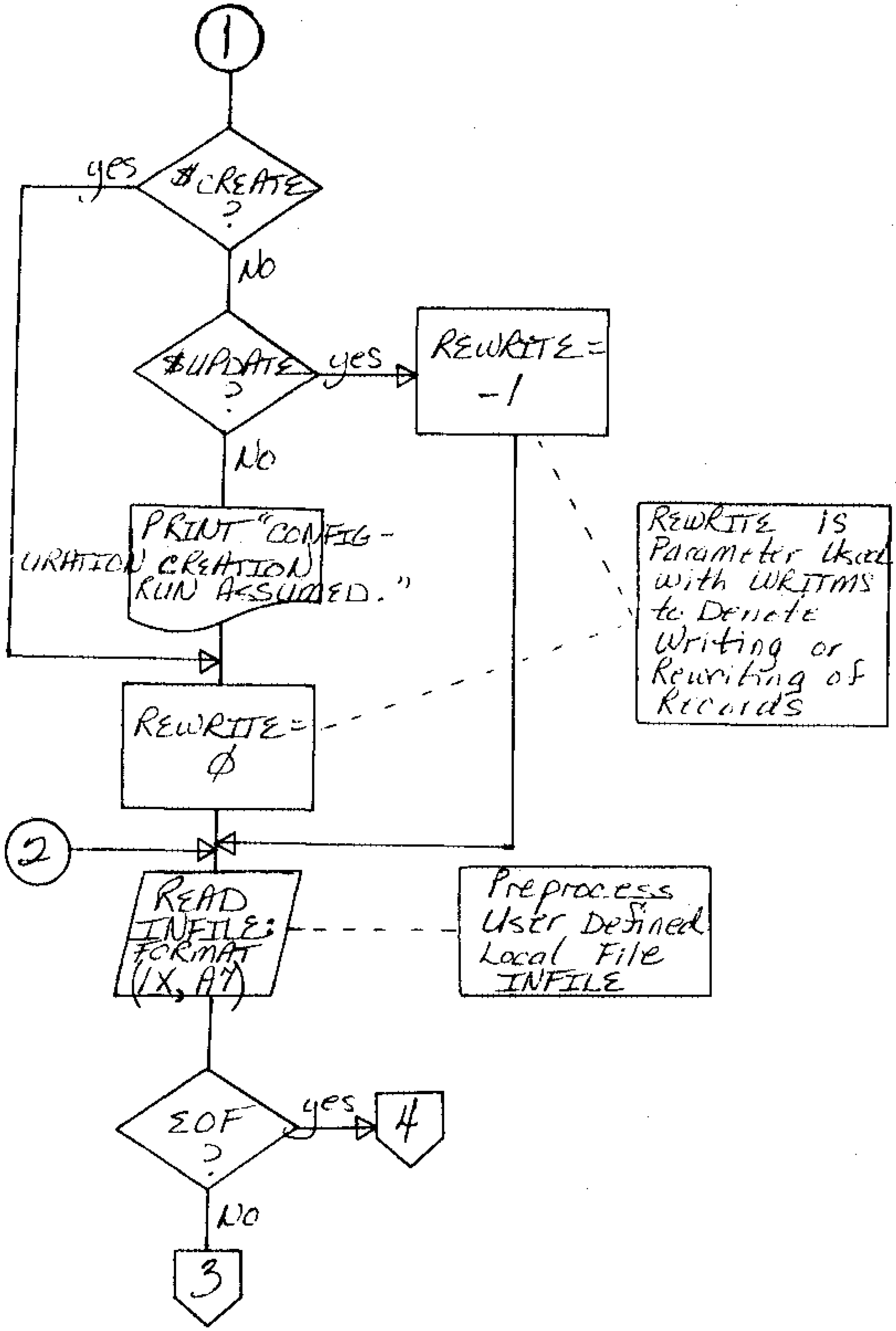


FORTRAN Library Routines: OPENM
CLOSMS
READMS
WRITMS

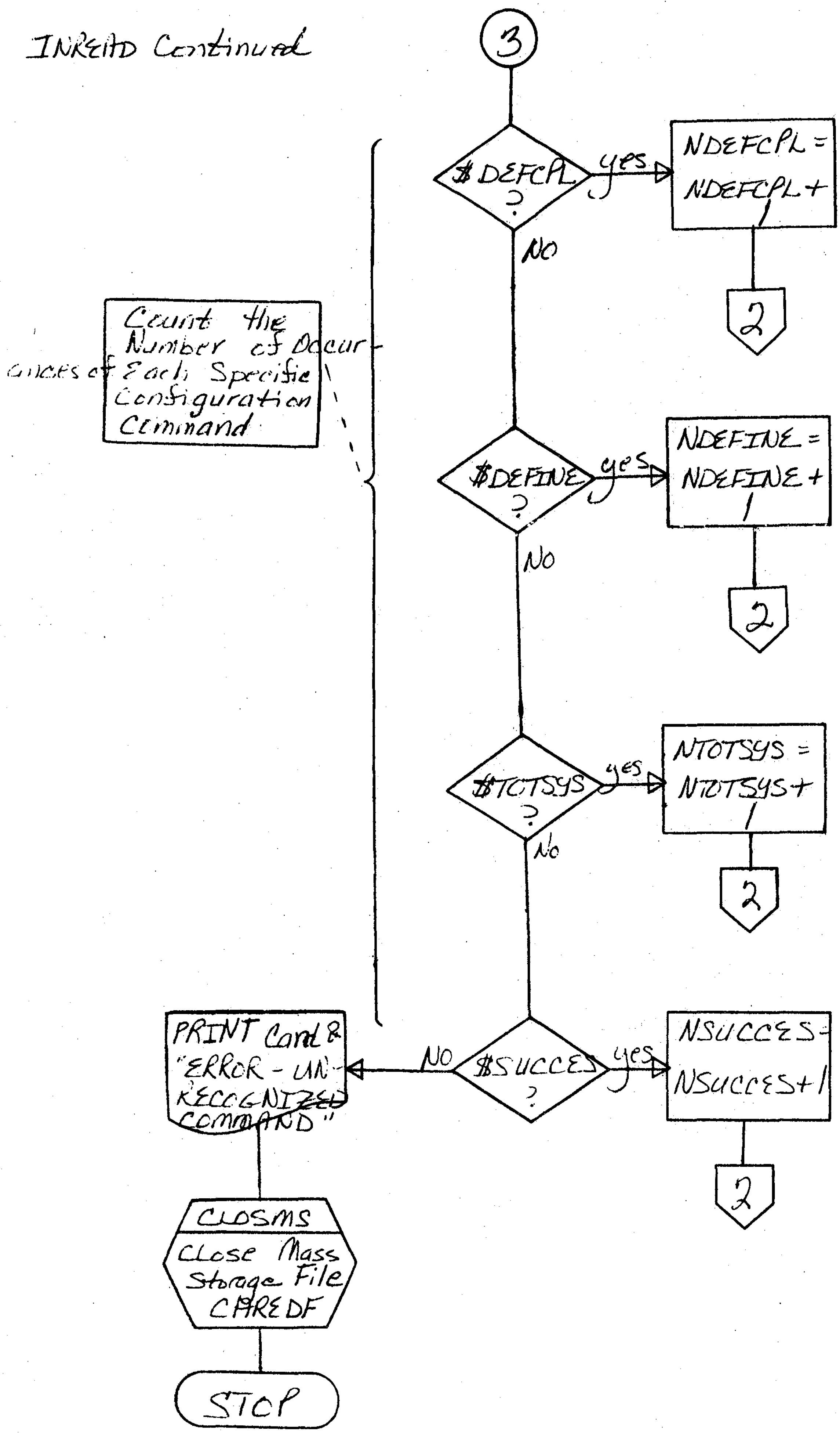


INREAD Subroutine Function Flow-Diagram

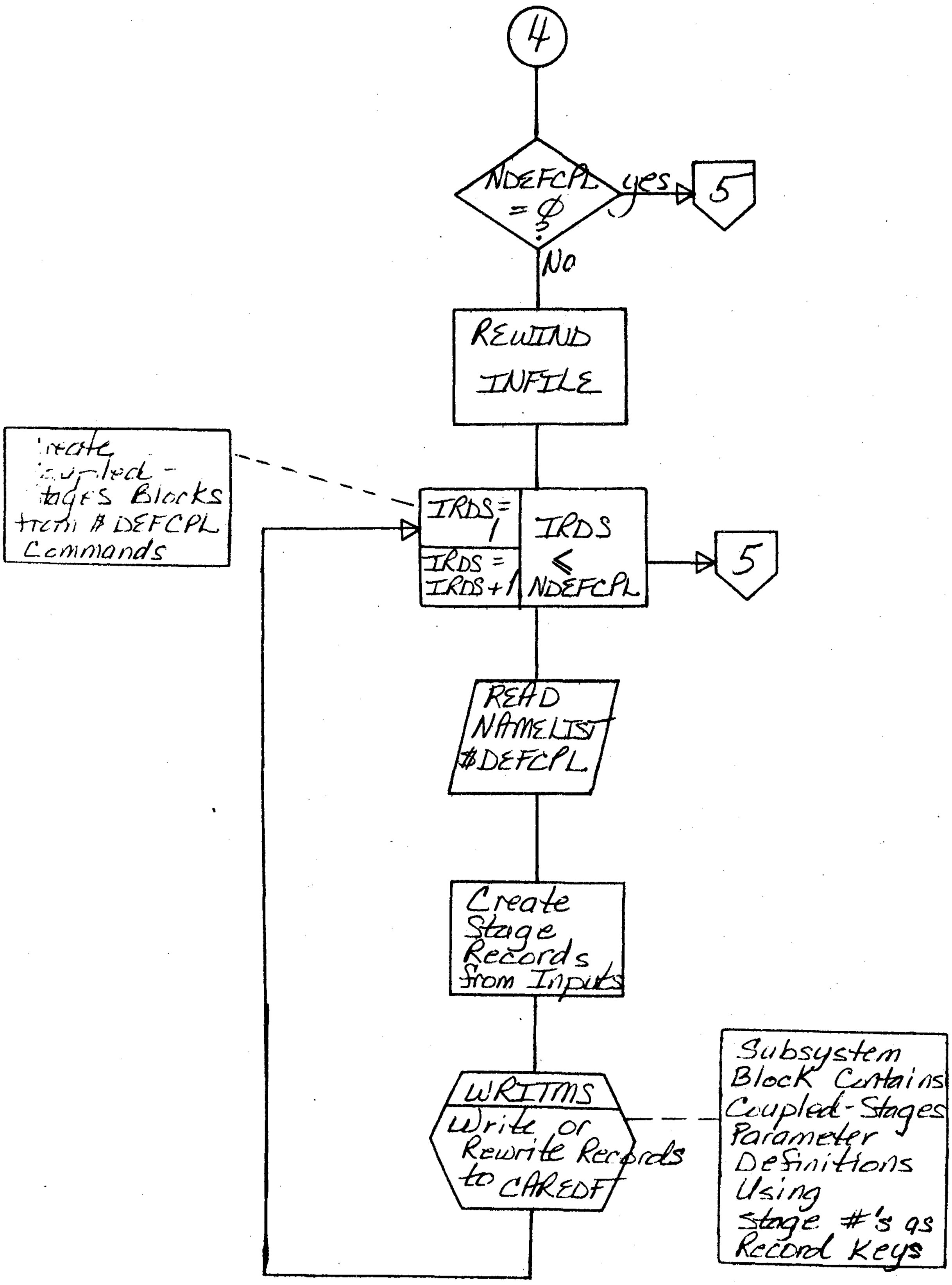
INREAD Continued



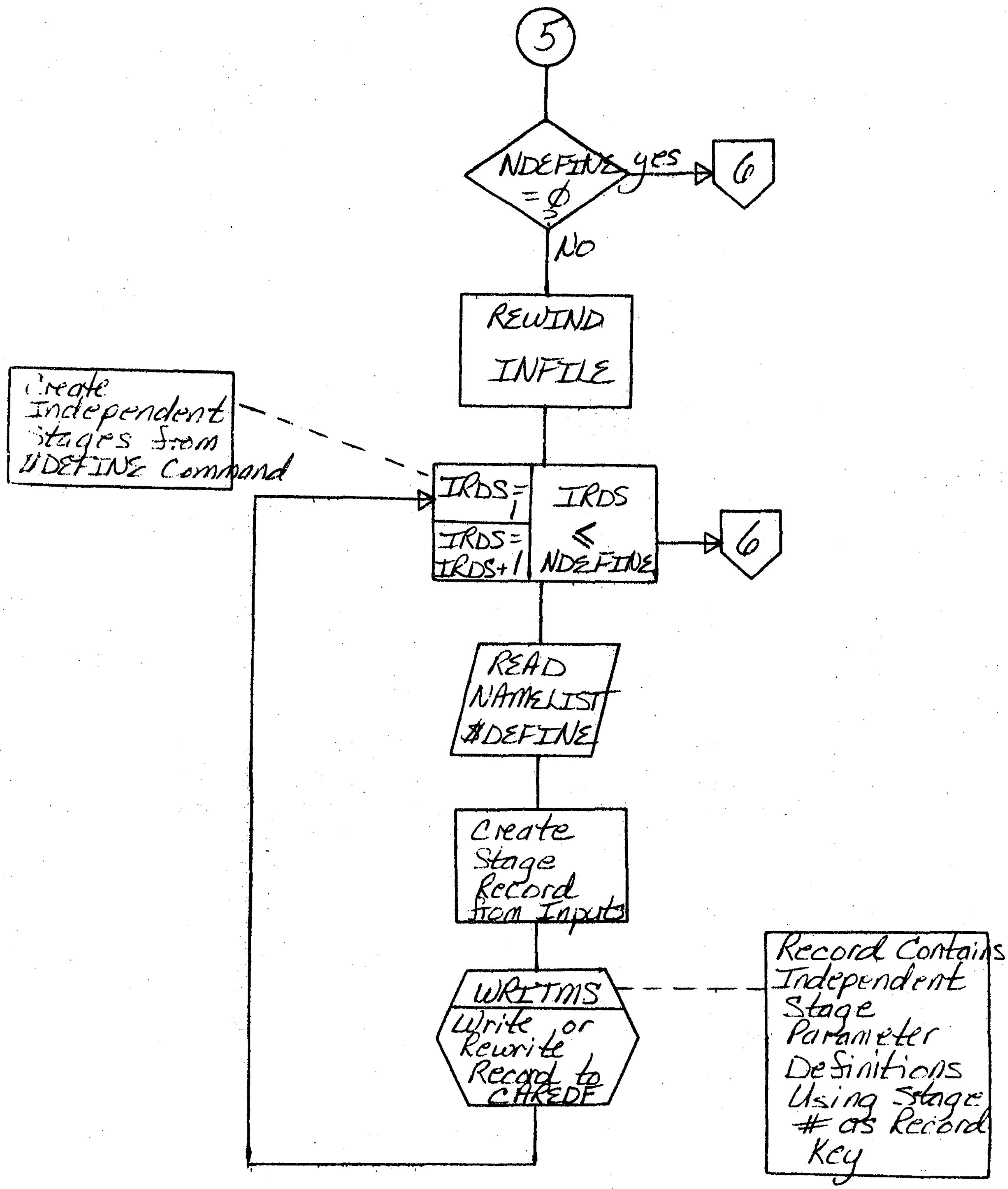
INREAD Continued



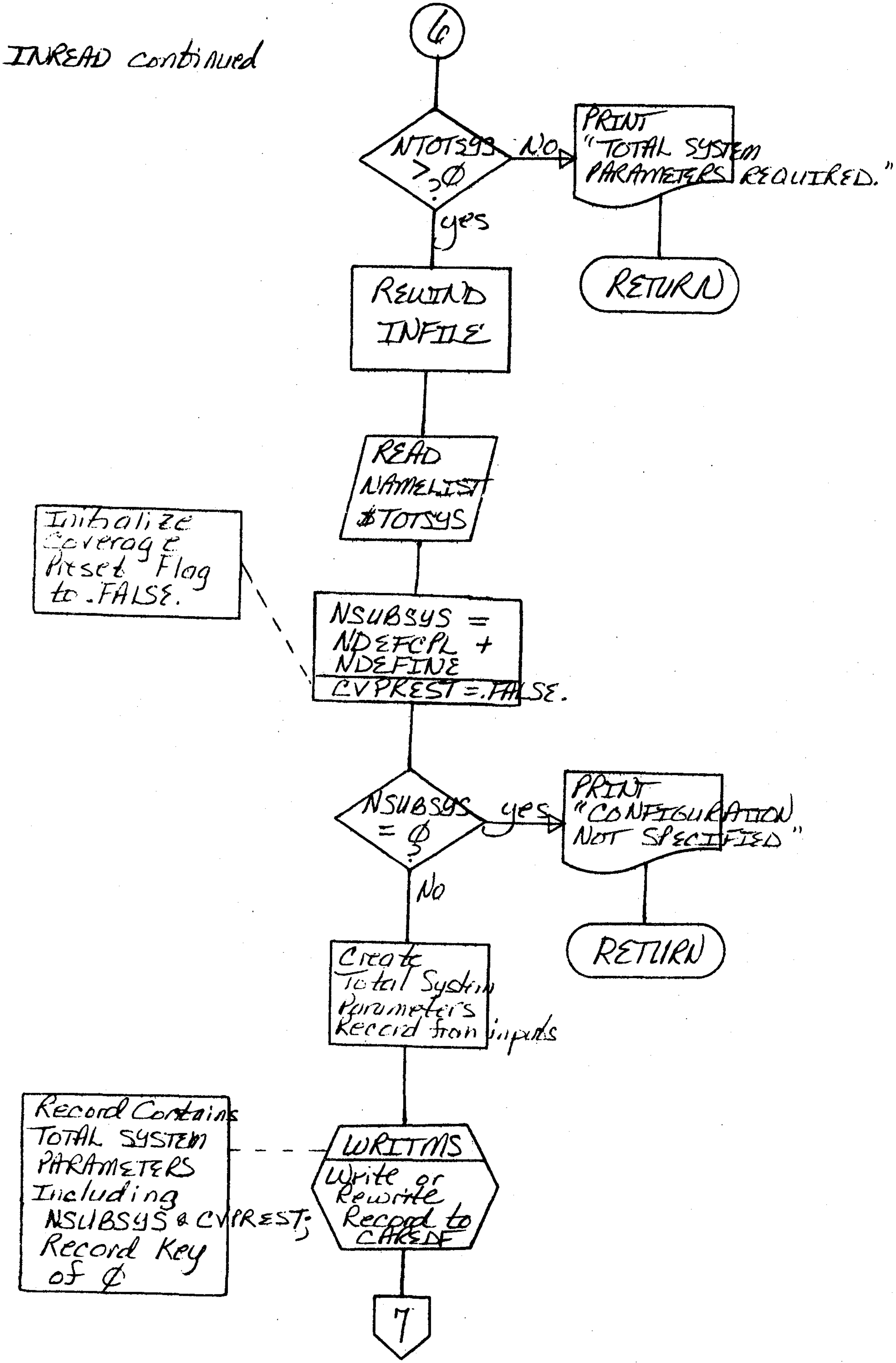
INREAD continued



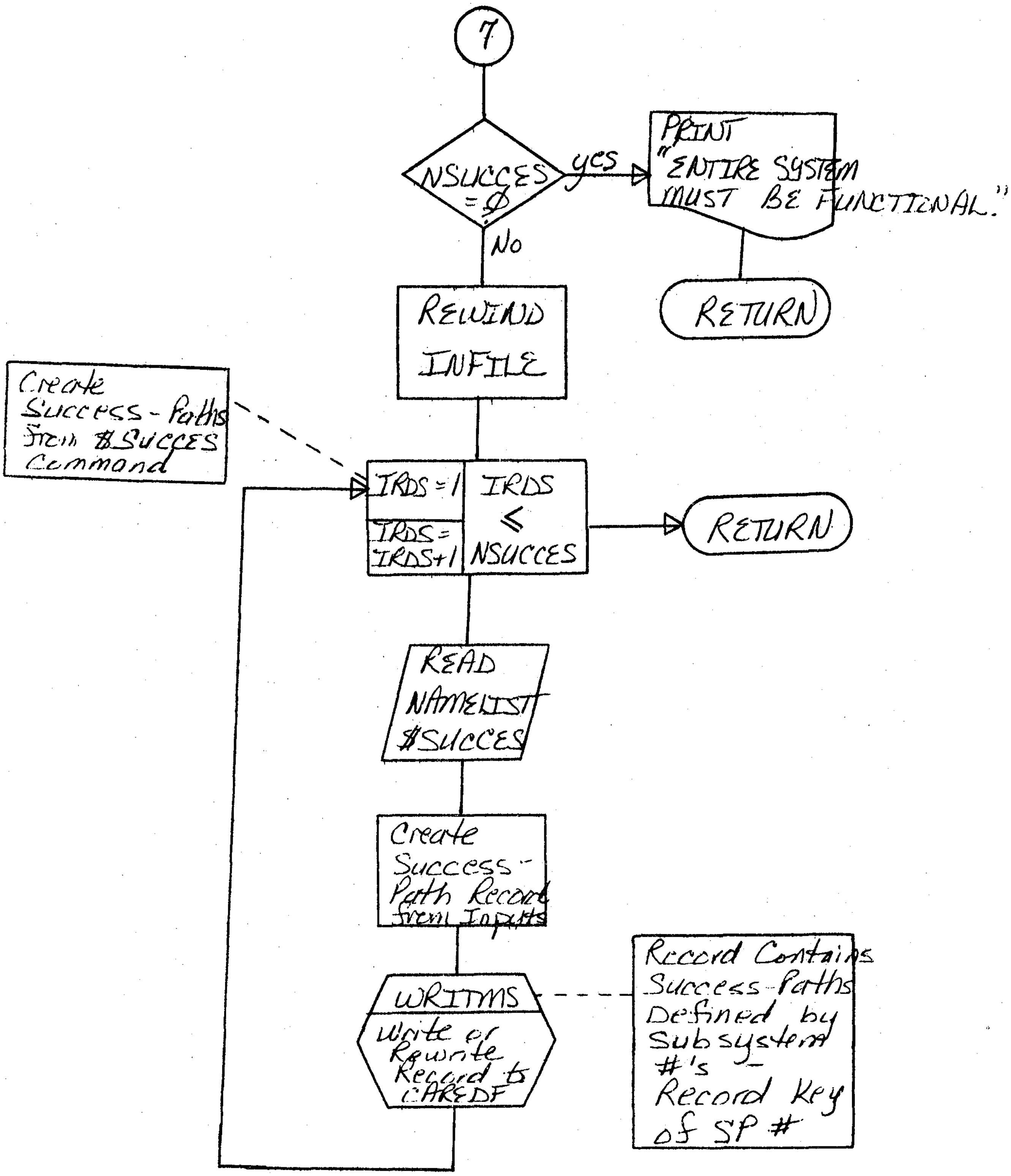
INREAD continued

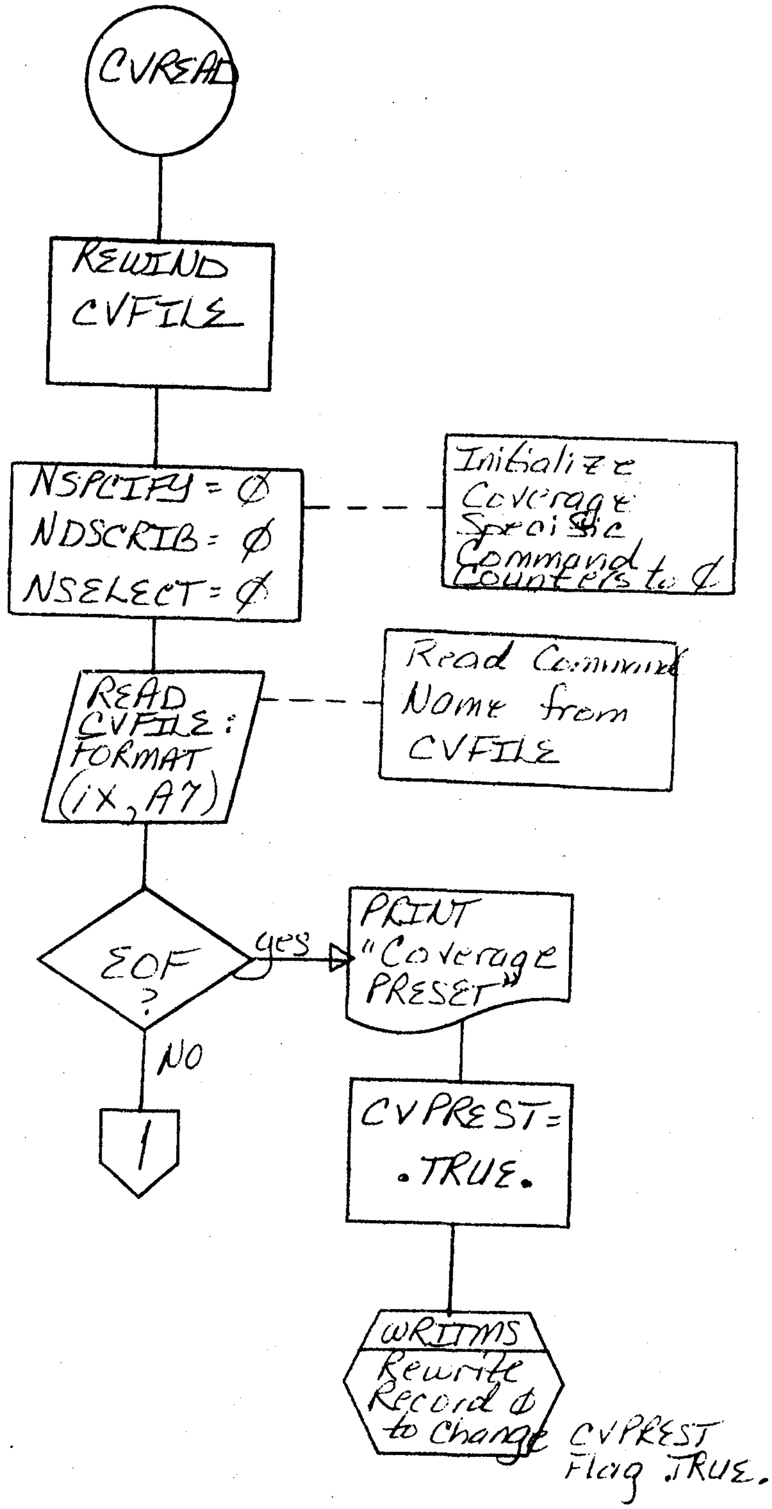


INREAD continued



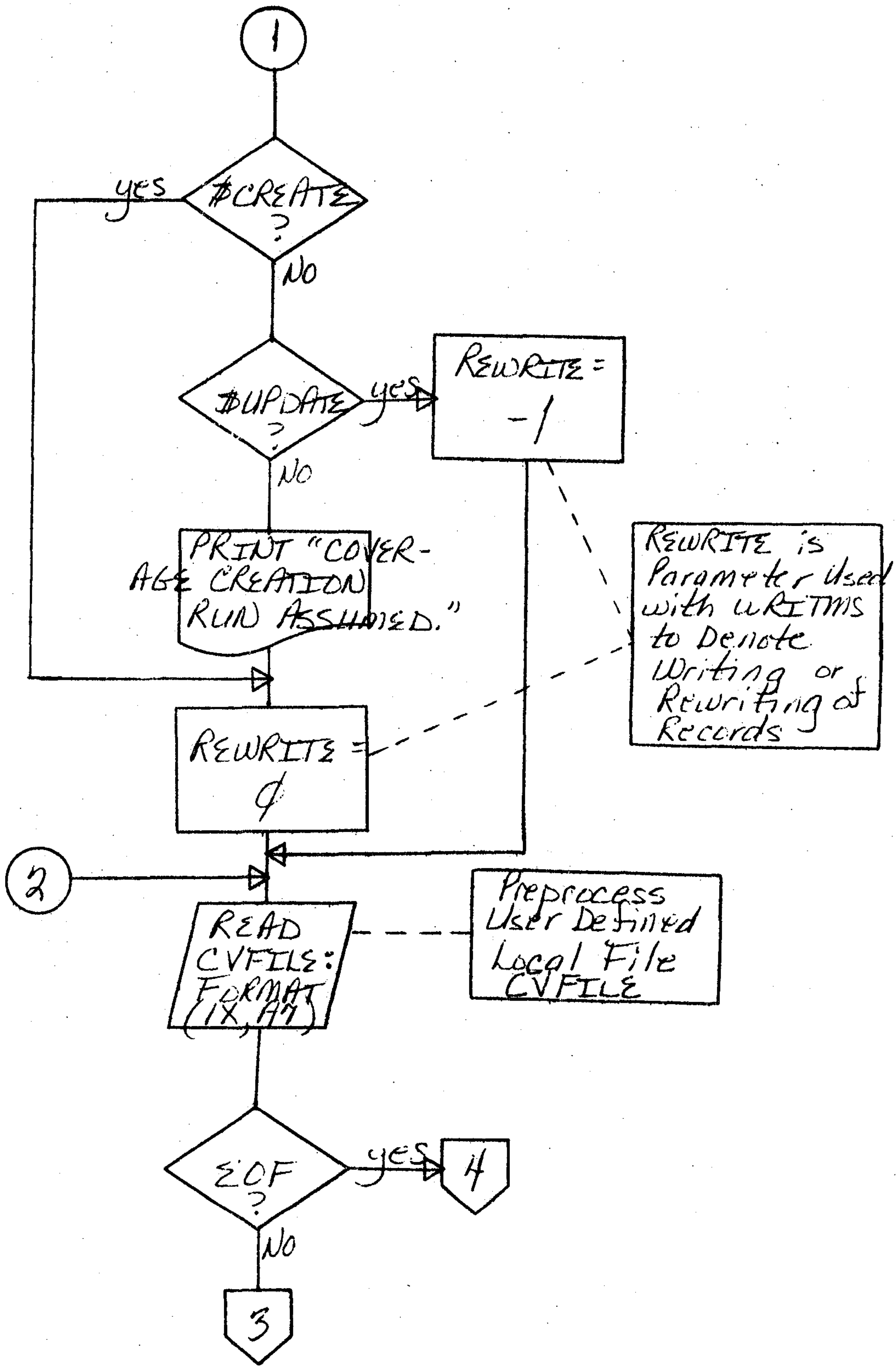
INKEHD Continued



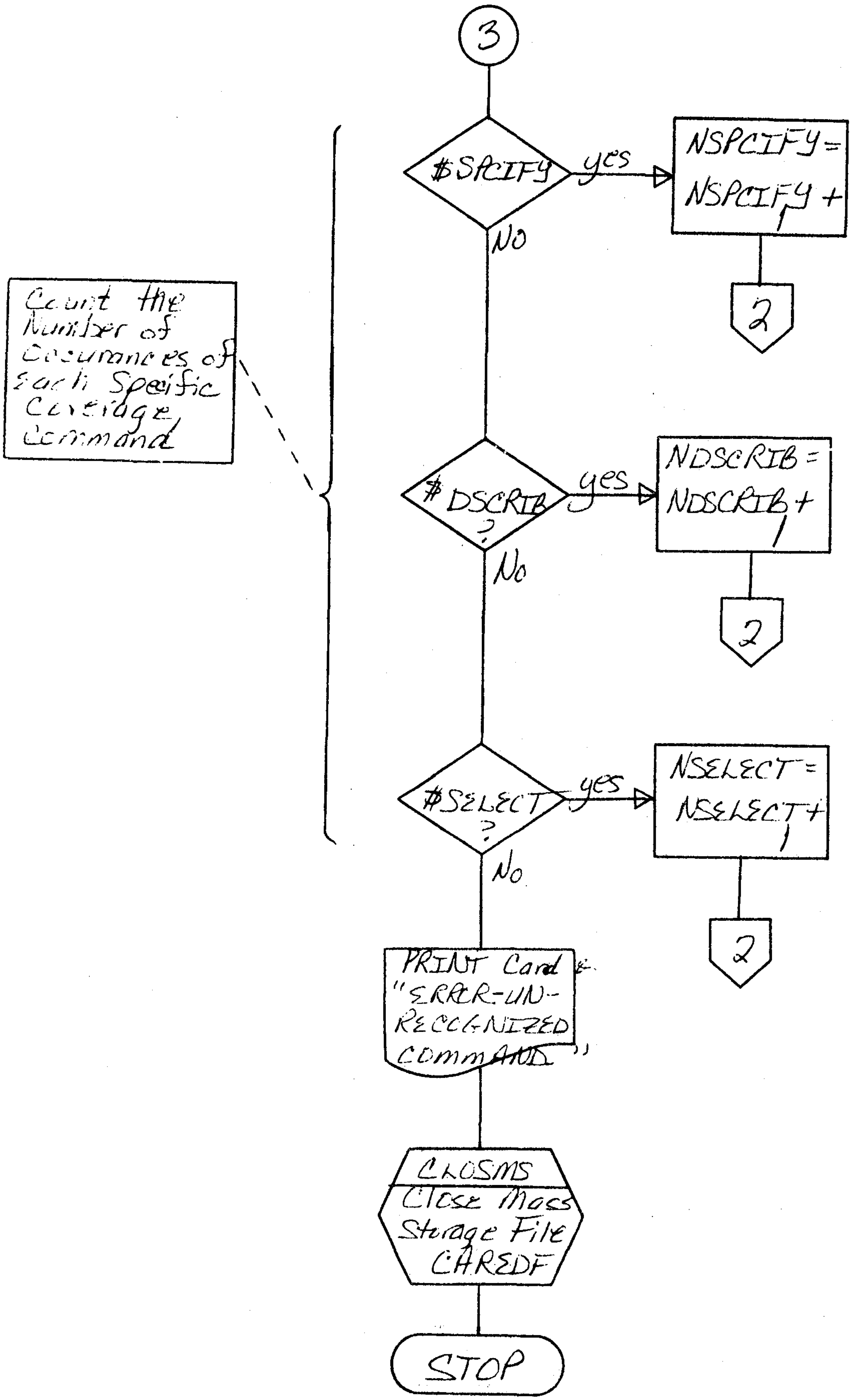


CVREAD Subroutine Functional Flow-Diagram

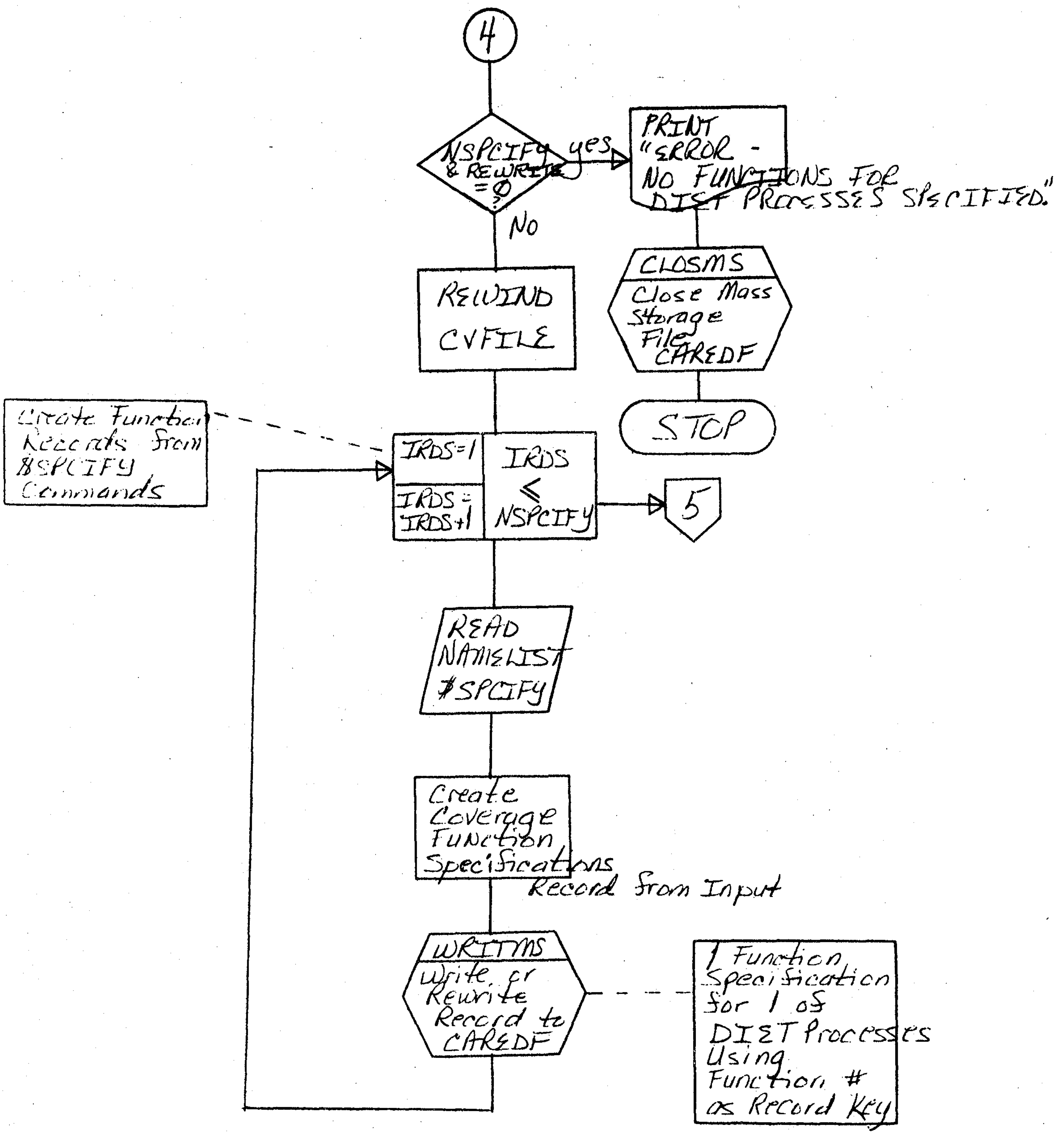
CVREAD Continued



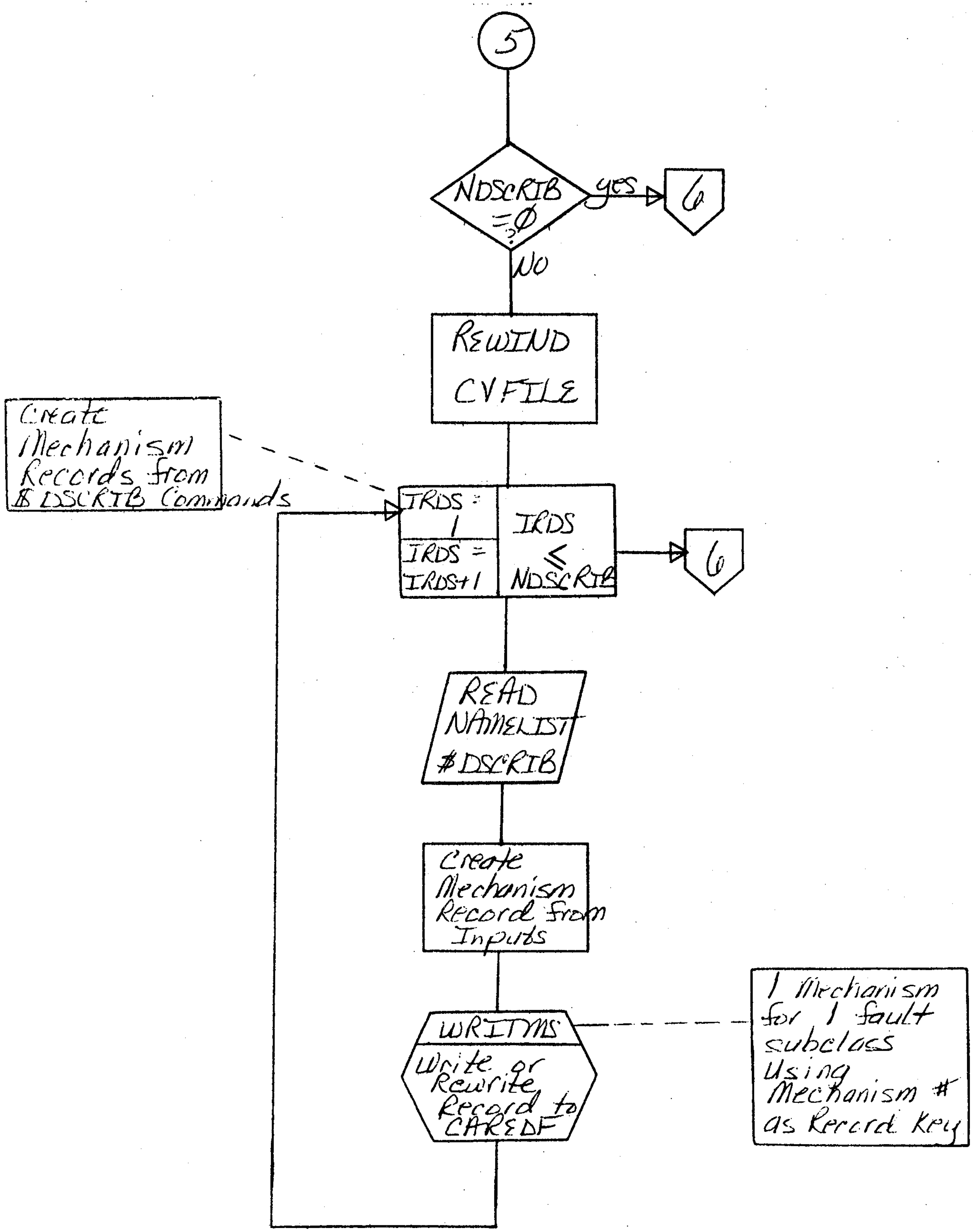
CVREAD continued



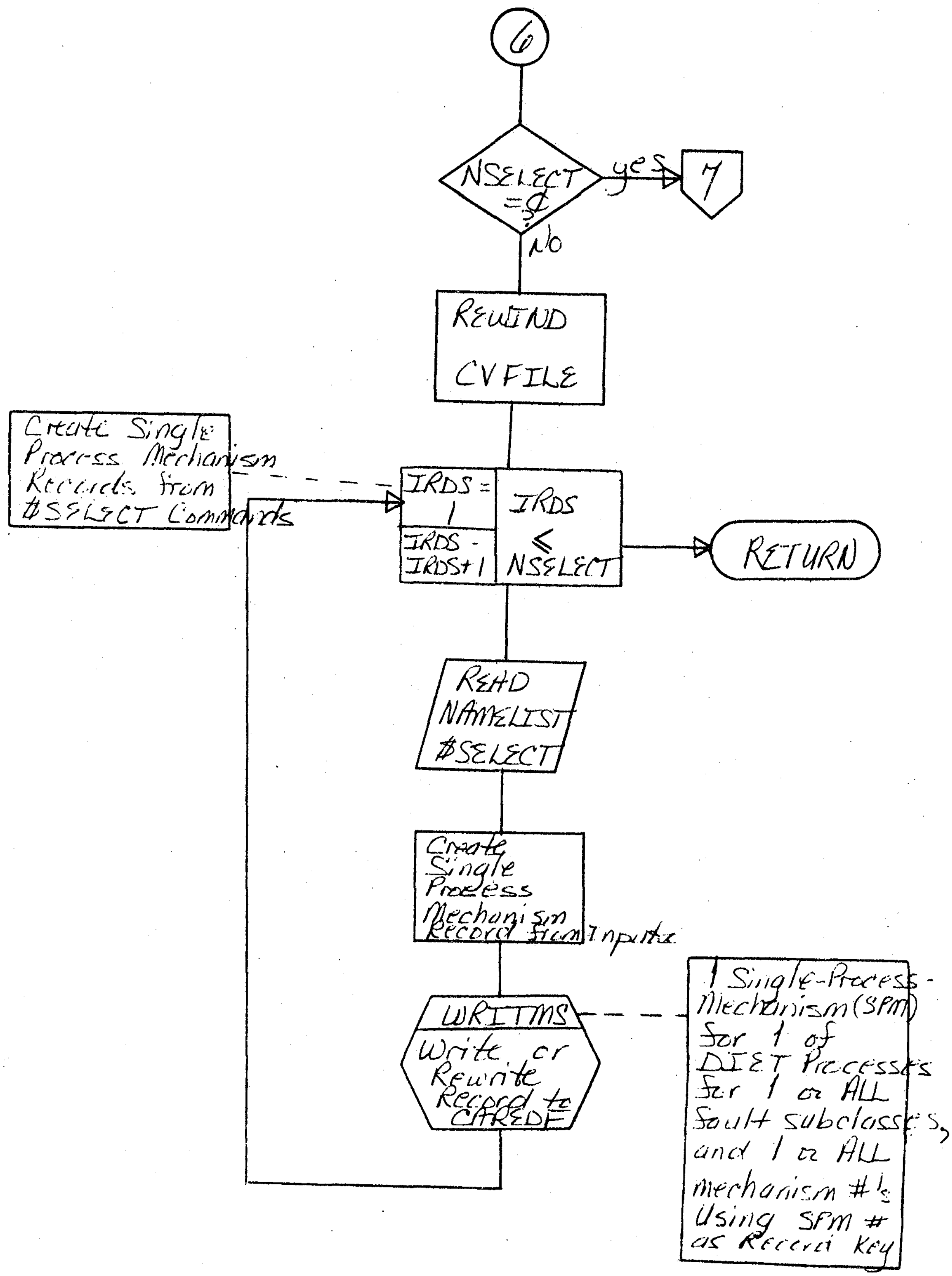
CVKEHD Continued



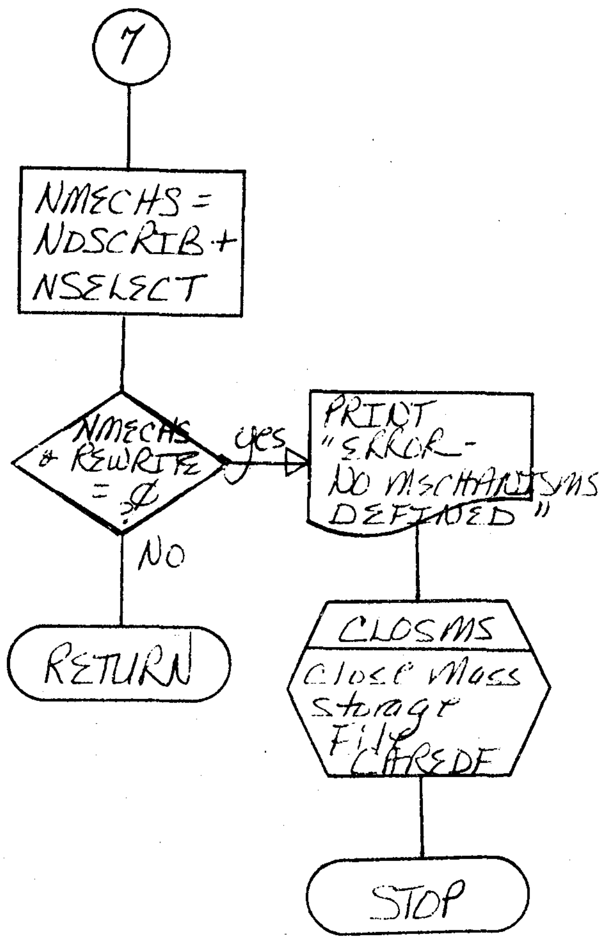
CVREAD Continued

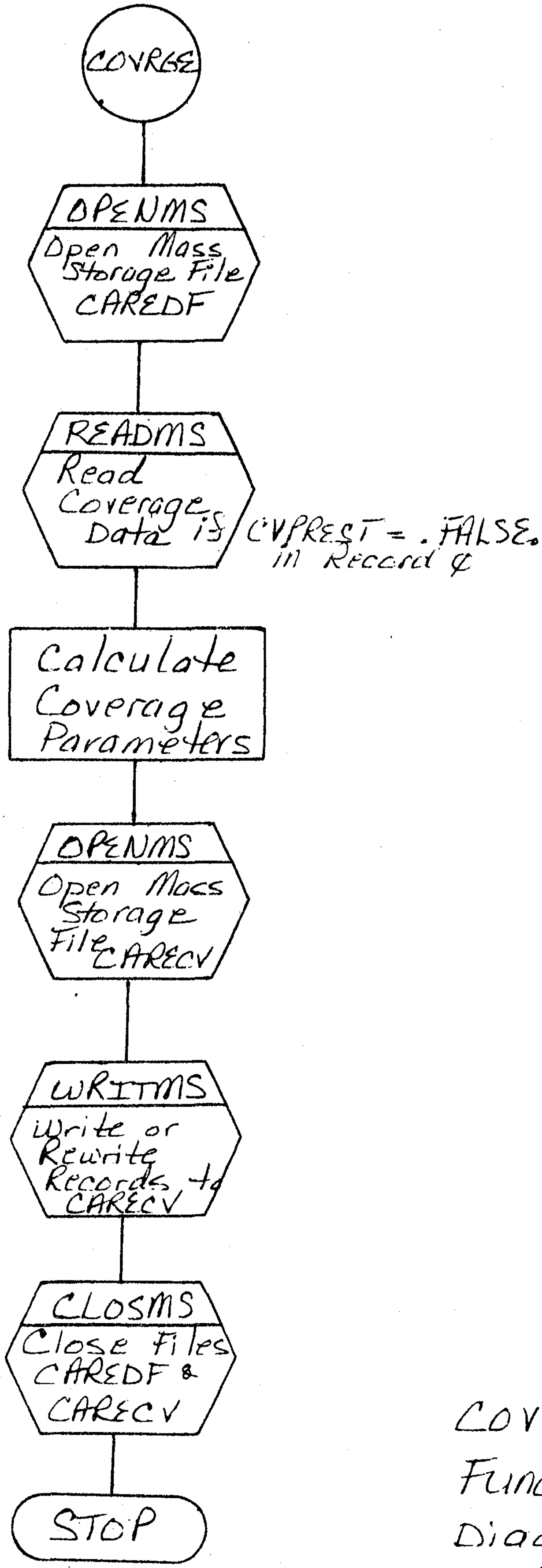


CVREAD Continued

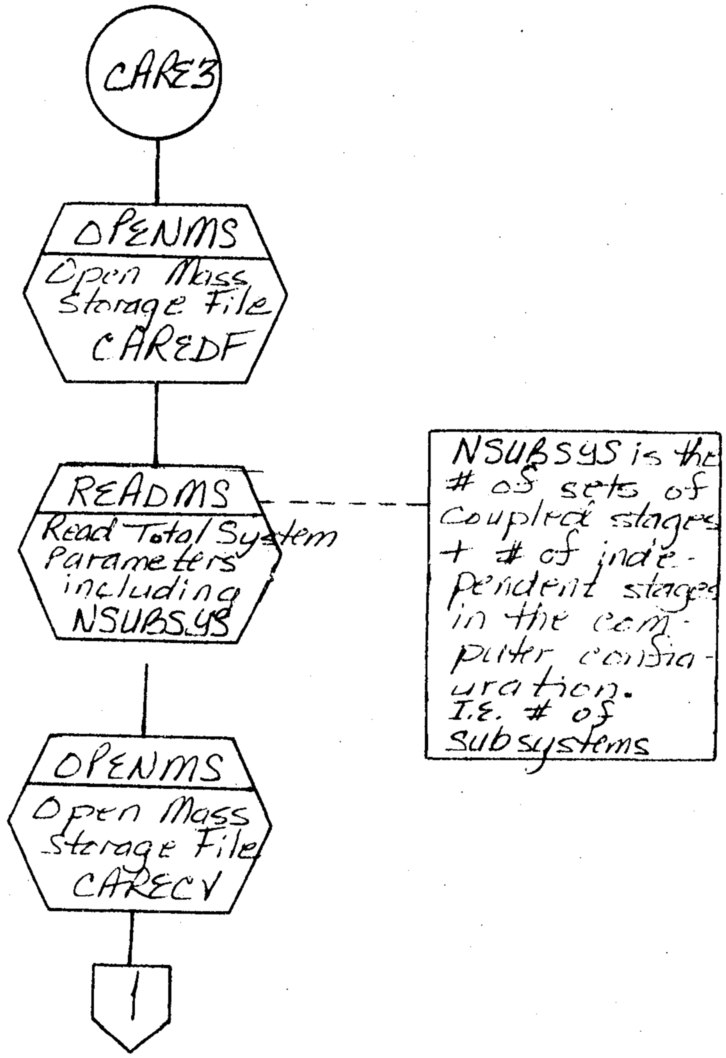


CVREHD Continued



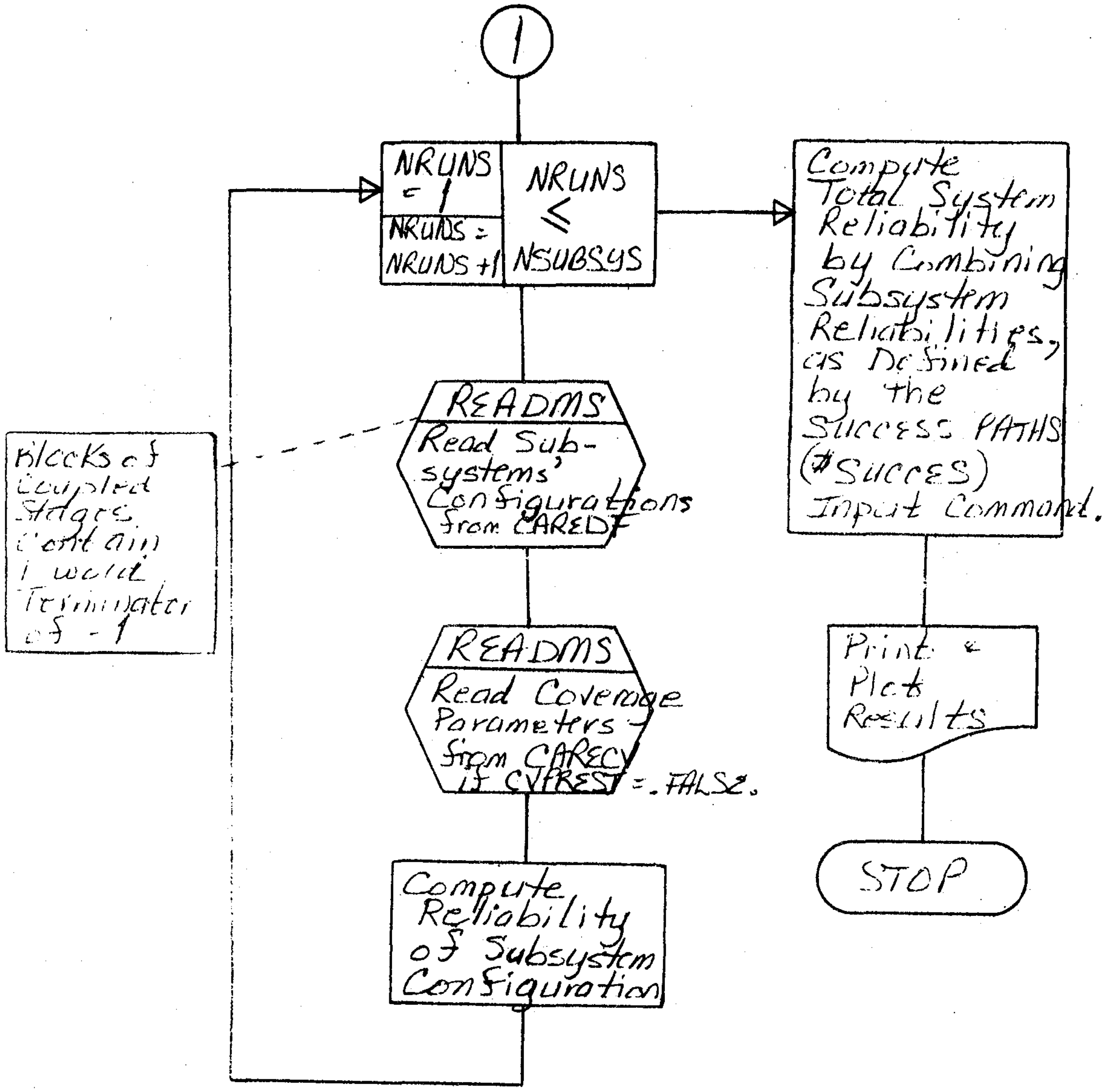


COVRGE Program Functional Flow Diagram



CARE3 Program Functional Flow-Diagram

CARE3 continued



Two sets of NAMELIST commands are necessary:

a. NAMELISTS to define the computer configuration and preset coverage data, if coverage is not to be calculated;

b. NAMELISTS to define necessary coverage data in order for routine COVRGE to compute coverage rates.

The basic NAMELISTS required are:

a1. $CREATE CAREDF or $UPDATE CAREDF
a2. $DEFCPL - define coupled-stages subsystem
a3. $DEFINE - define single stage subsystem
a4. $TOTSYS - define total system parameters
a5. $SUCCES - define success paths
b1. $CREATE CARECV or $UPDATE CARECV
b2. $SPCIFY - specify single process DIET functions
b3. $DSCRIB - describe mechanisms
b4. $SELECT - select functions for single-process DIET mechanisms

Other commands will also be necessary to choose output options, and more definition type commands may be added to the above list as the CARE III system evolves.

To specify the computer configuration that is to be modeled, NAMELIST's $DEFCPL and $DEFINE are used. Each define a subsystem of the entire configuration and are comprised of one or more stages. NAMELIST $TOTSYS defines the parameters necessary to describe the total system, and NAMELIST $SUCCES defines the required success paths of the total system. Also necessary is a NAMELIST like command which tells CAREIN whether this is a creation or update run. If $CREATE CAREDF is read as an input, the program generates a new random access mass storage

10-1349 (11 68) PRINTED IN U. S. A.

file using the previously mentioned NAMELIST commands. If $UPDATE CAREDF is read, an existing CAREDF file will be modified.

To specify a single DIET process of Detection, Isolation, Error-Propagation-Recovery, Time-Loss-Recovery processes, NAMELIST $SPCIFY is used. To describe mechanisms using the previously defined functions, NAMELIST's $DSCRIB and $SELECT are used. $DSCRIB describes a mechanism for all DIET processes for a given fault-subclass; $SELECT defines single-process DIET mechanisms for one or all fault-subclasses.

The following sample general variables will be required for each specified NAMELIST command:

1. NAMELIST $DEFCPL
   $DEFCPL SBSYSTM = Integer, NSTGES = Integer, ..., Integer, NUS = Integer, ..., Integer, NUSVS = Integer, ..., Integer, FLRTS = Real, ..., Real, CVPS = Reals, CVTS = Reals, CVIS = Reals$

This command defines a subsystem of the entire computer configuration made up of more than one stage, having the corresponding number of original units and number of required unit survivors. It also contains stage failure rates and preset coverage parameters if desired. For example, $DEFCPL SBSYSTM = 1, NSTGES = 1, 2, 3, NUS = 15, 9, 5, NUSVS = 2, 2, 2, FLRTS = 3*1.18E-4$

2. NAMELIST $DEFINE
   $DEFINE SBSYSTM = Integer, NSTG = Integer, NU = Integer, NUSV = Integer, FLRT = Real, CVP = Real, CVT = Real, CVI = Real$

This command defines a subsystem, comprised of an independent stage with its required parameters.

3. NAMELIST $TOTSYS

$TOTSYS STEP = Real, TMAX = Real, TBASE = Integer$

This command specifies the integration step size desired, the maximum time desired and the time base (1:hrs, 2:mins, 3:secs, 4:msecs).

4. NAMELIST $SUCCES

$SUCCES SP = Integer, PATHS = Integer, ..., Integer$

This command specifies the subsystem success paths to be used by routine CARE3 to compute the entire system's reliability.

5. NAMELIST $SPCIFY

$$\text{\$SPCIFY FUNC = Integer,} \left\{\begin{matrix} \text{D} = 1 \\ \text{I} = 1 \\ \text{E} = 1 \\ \text{T} = 1 \end{matrix}\right\}, \left\{\begin{matrix} \text{IMP} = 1 \\ \text{CON} = 1 \\ \text{PUL} = 1 \\ \text{EXP} = 1 \end{matrix}\right\},$$

$$\text{ISCH} = \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\}, \text{ IREP = Integer, INTF} = \left\{\begin{matrix} 0 \\ 1 \end{matrix}\right\},$$

COEF = Real, TDEL = Real, P1 = Real, P2 = Real, P3 = Real, TDUR = Real$*

This command specifies one function for one DIET process of the recovery system.

*NOTE: $\left\{ \right\}$ means "choose one of the enclosed variable definitions".

10-1349 (11 68) PRINTED IN U. S. A.

6. NAMELIST $DSCRIB

$DSCRIB MECH = Integer, FLTSUBC = Integer,

PRMFLTS = D#, I#, E#, T#,

TRNFLTS = D#, I#, E#, T#$*

This command describes one mechanism for one fault subclass using the previously defined functions.

7. NAMELIST $SELECT

$SELECT FUNC = Integer, SPM = Integer,

$$\left\{\begin{array}{l} D = 1 \\ I = 1 \\ E = 1 \\ T = 1 \end{array}\right\} , \text{ FLTSUBC } = \left\{\begin{array}{l} \text{Integer} \\ >8 \text{ for ALL} \end{array}\right\} ,$$

$$\text{ERRTYP} = \left\{\begin{array}{l} 1 \text{ (permanent)} \\ 2 \text{ (transient)} \\ 3 \text{ (both)} \end{array}\right\} ,$$

$$\text{MECH} = \left\{\begin{array}{l} \text{Integer} \\ >20 \text{ for ALL} \end{array}\right\} \$$$

This command selects previously defined functions for single-process mechanism(s) for fault subclass(es).

For further illustration of the proposed NAMELIST as template input scheme, the system defined in Chart 3-2 would be defined using the following commands: (Note: Coverage is not being preset.)

```
$CREATE CAREDF
$DEFCPL SBSYSTM = 1, NSTGES = 1, 2, 3, NUS = 15, 9, 5,
        NUSVS = 2, 2, 2, FLRTS = 3*1.18E-4$
$DEFCPL SBSYSTM = 2, NSTGES = 4, 5, 6, 7, NUS = 7, 5, 5, 4,
        NUSVS = 3, 2, 1, 2, FLRTS = 4*1.0E-4$
$DEFCPL SBSYSTM  = 3, NSTGES = 8, 9, 10, 11, 12, NUS = 2, 3, 4,
        3, 3, NUSVS = 1, 2, 2, 1, 1, FLRTS = 3*1.18E-4,
        2*1.0E-4$
```

*# represents a previously defined function number.

```
$DEFCPL SBSYSTM = 4, NSTGES = 13, 14, 15, NUS = 8, 7, 6,
        NUSVS = 3, 2, 2, FLRTS = 3*1.18E-4$
$DEFCPL SBSYSTM = 5, NSTGES = 16, 17, NUS = 20, 15,
        NUSVS = 5, 3, FLRTS = 2*0.95E-4$
$DEFINE SBSYSTM = 6, NSTG = 18, NU = 10, NUSV = 5, FLRT =
        1.0E-4$
$DEFINE SBSYSTM = 7, NSTG = 19, NU = 8, NUSV = 3, FLRT =
        1.18E-4$
$DEFINE SBSYSTM = 8, NSTG = 20, NU = 5, NUSV = 2, FLRT =
        1.5E-4$
$TOTSYS STEP = 2.0, TMAX = 1000.0, TBASE = 1$
$SUCCES SP = 1, PATHS = 1, 3, 5, 8$
$SUCCES SP = 2, PATHS = 2, 4, 6, 7, 8$
```

To make an update run, if certain changes are desired in some of the stages, the input stream could look as follows:

```
$UPDATE CAREDF
$DEFCPL SBSYSTM = 3, NSTGES = 10, 12, NUS = 3, 2$
$DEFINE SBSYSTM = 7, NSTG = 19, NUSV = 4, FLRT = 1.0E-4$
```

Only those parameters that are to be changed need to be listed because of the nature of NAMELIST commands. If a parameter need not be changed, it remains as previously defined.

3.3.2 COVERAGE CALCULATOR

Program COVRGE will not require any direct user input. Its input will be supplied by the Direct Access Information System (DAIS) file CAREDF generated by the input processor routine CAREIN. If coverage parameters are to be calculated by program COVRGE, records must exist within file CAREDF which describe all necessary recovery functions and D/I/R mechanisms. This corresponds to the coverage model which exists in CARE II. Also contained in this file are records describing the

10-1349 (11 68) PRINTED IN U. S. A.

intermittent coverage model (see Section 4.2.3 of the CARE III Final Report, Phase 1).

Program COVRGE will calculate all necessary coverage parameters and write these parameters to DAIS file CARECV for processing by program CARE3.

The following diagram depicts the user's options when defining the coverage model desired.

3.3.3 CARE3 RELIABILITY ESTIMATOR

Program CARE3 will read DAIS files CAREDF and CARECV and compute the reliability of the configuration specified within these files. If N subsystems were defined, N iterations of the reliability computation section of CARE3 will be performed. CARE3 will then compute the total system reliability using the success paths specified by the user. (See CARE III Final Report for details on the mathematical reliability model chosen for the CARE III system.)

The complexity of the output generated by CARE3 will depend upon user commands input to CAREIN.

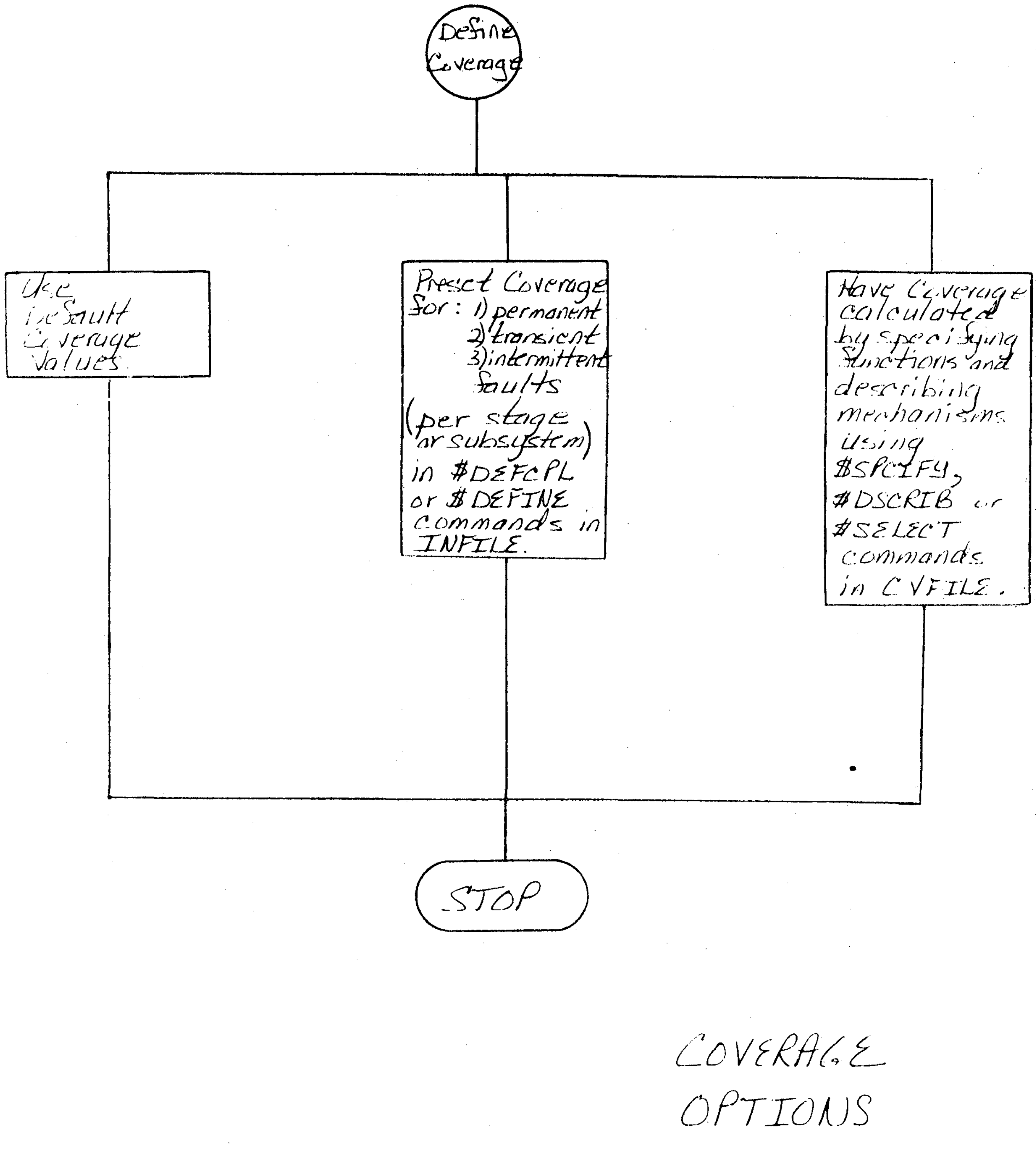

Define Coverage
Use Default Coverage Values
Preset Coverage for: 1) permanent 2) transient 3) intermittent faults (per stage or subsystem) in #DEFCPL or $DEFINE commands in INFILE.
Have Coverage calculated by specifying functions and describing mechanisms using $SPCIFY, #DSCRIB or #SELECT commands in CVFILE.
STOP
COVERAGE OPTIONS

Diagram 3.3-1

4.0 QUALITY ASSURANCE

4.1 INTRODUCTION

An acceptable test sequence will be written to test each subprogram, program and system as a whole.

4.2 TEST REQUIREMENTS

The accuracy of the system will be displayed by modeling systems with published assessment results and then comparing these to the CARE III results. In addition, configurations will be postulated that can be treated analytically but that exercise significant portions of the CARE III program, thereby allowing CARE III results to be compared with analytically derived results. Finally, sensitivity analyses will be conducted to verify that small parametric deviations produce appropriate deviations in the results. Where possible, checks on the magnitude of these deviations will be made analytically using, for example, power series expansions.

4.3 ACCEPTANCE TEST REQUIREMENTS

The CDC file management control statement TDUMP will be used to list the DAIS files to be certain that the files contain the proper data in the format expected by programs COVRGE and CARE3.

A TDUMP listing shall accompany the delivered test sequence to further illustrate the internal workings of the CARE III system.

10-1349 (11 68) PRINTED IN U. S. A.

APPENDIX 1

MATRIX METHODS FOR MARKOV MODEL SOLUTIONS

A1.1 INTRODUCTION

A1.1.1 STATEMENT OF PROBLEM

The Markov model defines a system of n first-order differential equations of the form

$$X' = AX, \qquad (*)$$

where A is an n x n transition matrix, and

$$X(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \cdot \\ \cdot \\ \cdot \\ x_n(t) \end{pmatrix};$$

$x_i(t)$ is the probability that the system is in state i at time t.

If we could assume that A is diagonalizable, then there exist k distinct eigenvalues ($k \leq n$) of A each with algebraic multiplicity m_i ($i = 1, 2, \ldots, k$) such that

$$\sum_{i=1}^{k} m_i = n,$$

and corresponding to each eigenvalue λ_i there are m_i independent eigenvectors $X_{ij} = (j = 1, 2, \ldots, m_i)$. Thus, we can assume that a fundamental set of solutions of (*) can be found that has the following form:

$$S = \left\{ X_{11}e^{\lambda_1 t}, \ldots, X_{1m_1}e^{\lambda_1 t}, X_{21}e^{\lambda_2 t}, \ldots, X_{2m_2}e^{\lambda_2 t}, \ldots, \right.$$

$$\left. X_{k1}e^{\lambda_k t}, X_{km_k}e^{\lambda_k t} \right\}.$$

Therefore, S consists of n independent solutions of (*), and any solution of (*) can be written as a linear combination of the elements of S. Obviously, if all the eigenvalues and their corresponding independent eigenvectors can be found, a fundamental set of solutions of (*) can be constructed.

In general, however, A cannot be assumed to be diagonalizable. We can still find a fundamental set of solutions of (*),

$$S' = \left\{ X_1(t), X_2(t), \ldots, X_n(t) \right\},$$

where $X_i(t)$ does not, in general, assume the simple form

expressed for the vectors of S. We state briefly that for each eigenvector λ_i with algebraic multiplicity m_i, there exist m_i independent solutions of (*), say $X_{ij}(t)$ $(j = 1, 2, \ldots, m_i)$, such that

$(A - \lambda_i I)^{m_{ij}} X_{ij} \equiv 0$, for $m_{ij} \leq m_i$ (but for no $m < m_{ij}$);

$X_{ij}(t)$ has the form $X_{ij}(t) = (p_{i1}(t)e^{\lambda_i t}, \ldots, p_{in}(t)e^{\lambda_i t})$, (A)

where $p_{ik}(t)$ $(k = 1, 2, \ldots, n)$ are polynomials of degree $\leq m_{ij} - 1$. Thus, $X_{ij}(t)$ makes up the fundamental set S'; they are called primitive solutions of (*).

Thus, the major problem that confronts us is one of finding all λ and all independent vectors X such that

$$AX = \lambda X \tag{**}$$

A1.1.2 BRIEF OUTLINE OF REPORT

For general square matrices, the eigenproblem is probably best approached by means of matrix similarity transformations. Initially, we shall assume that A is a completely general square matrix, and in this connection the Givens' method, which reduces A to lower Hessenberg form, is presented (see section A1.2.1). Sections A1.2.2 and A1.2.3 introduce Hyman's theorem and the Newton-Raphson method, respectively. The Hyman theorem is used to evaluate the characteristic polynomial (of the Hessenberg matrix) and its first derivative while Newton's method actually computes the eigenvalues. The computation of eigenvectors takes place in A1.3, where Gaussian elimination and the method of inverse iteration can be found.

In A1.5 and A1.6, we assume that A is diagonalizable. Here we provide some of the well-known theorems on well-posedness and a posteriori error estimates.

A1.7 gives an algorithm for computing the eigenvalues of large sparce matrices. The highlights of this section, in which sparcity is used heavily, are sparce Gaussian elimination and the Laguerre iteration technique. It is hoped that this section will be the basis of a computer program to solve large sparce systems.

In sections A1.4 and A1.7.9, a priori error estimates are given for the Givens' method and the Gaussian reduction scheme, respectively.

A1.2 AN ALGORITHM FOR GENERAL MATRICES

A1.2.1 GIVENS' METHOD

We now consider a method involving matrix transformations to reduce the matrix A to lower Hessenberg form. The matrix $A \equiv (a_{ij})$ is in lower Hessenberg form if and only if $a_{is} = 0$ for $i + 2 \leq s \leq n$. That is, every element above the upper codiagonal elements is zero. Once we have obtained this reduced form, a method due to Hyman can be used to evaluate the characteristic polynomial of A and thereby help us to compute the eigenvalues.

We shall introduce the algorithm of Givens with a 4 x 4 matrix A and then work our way up to the general algorithm.

First, let

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

To reduce the matrix A to lower Hessenberg form, we must annihilate the elements a_{13}, a_{14} and a_{24}. This can be accomplished by constructing the finite sequence of matrices $\{P_k\}$, k = 1, 2,..., M, and then define

$$B_0 \equiv A,$$
$$B_k \equiv P_k^* B_{k-1} P_k, \qquad 1 \le k \le M. \tag{1}$$

(P^* is the conjugate transpose of the matrix P.)

To annihilate the element a_{13}, we let the 4 x 4 matrix $P_1 \equiv \left(p_{ij}^{(1)}\right)$ be such that

$$p_{22}^{(1)} = p_{33}^{(1)} = \frac{a_{12}}{\sqrt{a_{12}^2 + a_{13}^2}},$$

$$p_{23}^{(1)} = -p_{32}^{(1)} = \frac{-a_{13}}{\sqrt{a_{12}^2 + a_{13}^2}}, \tag{2}$$

$$p_{rs}^{(1)} = \delta_{rs} \quad \text{for all other } (r, s).$$

Therefore, with $p_{ij}^{(1)}$ replaced by p_{ij}

$$B_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p_{22} & p_{32} & 0 \\ 0 & p_{23} & p_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p_{22} & p_{23} & 0 \\ 0 & p_{32} & p_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ p_{22}a_{21}+p_{32}a_{31} & p_{22}a_{22}+p_{32}a_{32} & p_{22}a_{23}+p_{23}a_{33} & p_{22}a_{24}+p_{32}a_{34} \\ p_{23}a_{21}+p_{33}a_{31} & p_{23}a_{22}+p_{33}a_{32} & p_{23}a_{23}+p_{33}a_{33} & p_{23}a_{24}+p_{33}a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \text{x}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p_{22} & p_{23} & 0 \\ 0 & p_{32} & p_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} & a_{12}p_{22}+a_{13}p_{33} & a_{12}p_{23}+a_{13}p_{33} & a_{14} \\ p_{22}a_{21}+p_{32}a_{31} & p_{22}(0)+p_{32}(00) & p_{23}(*)+p_{33}(**) & p_{22}a_{24}+p_{33}a_{34} \\ p_{23}a_{21}+p_{33}a_{31} & p_{22}(\triangle)+p_{32}(\triangle\triangle) & p_{23}(\square)+p_{33}(\square\square) & p_{23}a_{24}+p_{33}a_{34} \\ a_{41} & a_{42}p_{22}+a_{43}p_{32} & a_{42}p_{23}+a_{43}p_{33} & a_{44} \end{pmatrix}$$

where $* = p_{22}a_{22}+p_{32}a_{32}$, $** = p_{22}a_{22}+p_{32}a_{33}$, $\square = p_{23}a_{22}+p_{33}a_{32}$,

$\square\square = p_{23}a_{23}+p_{33}a_{33}$, $0 = p_{22}a_{22}+p_{32}a_{32}$, $00 = p_{22}a_{23}+p_{32}a_{33}$,

$\triangle = p_{23}a_{22}+p_{33}a_{32}$, and $\triangle\triangle = p_{23}a_{23}+p_{33}a_{33}$.

If we look at the element in the first row and third column of B_1, we see that our goal has been met. That is, from (2) we get

$$a_{12}p_{23}^{(1)} + a_{13}p_{33}^{(1)} = a_{12}\left(\frac{-a_{13}}{\sqrt{a_{12}^2 + a_{13}^2}}\right) + a_{13}\left(\frac{a_{12}}{\sqrt{a_{12}^2 + a_{13}^2}}\right) = 0$$

This process has to be used a total of three times to reduce A, for $n = 4$, to lower Hessenberg form. In general, the process must be carried out M times, where $M = \frac{(n-2)(n-1)}{2}$. It is easy to see that $M = 3$ when $n = 4$.

To formulate a general theorem for the Givens' method, we list in sequence the indices of the elements to be annihilated.

$$(1, 3), (1, 4), \ldots (1, n), (2, 4), (2, 5), \ldots, (2, n), \ldots, (n-2, n) \tag{4}$$

THEOREM (Givens): Let A be a real n x n matrix; let $B_0 \equiv A$. Let $(i-1, j)$ be the kth pair of indices in the sequence (4), and B_{k-1} have elements $(b_{rs}^{(k-1)})$. Let $P_k \equiv I$, if $b_{i-1,j}^{(k-1)} = 0$; otherwise, let $P_k \equiv (p_{rs}^{(k)})$ where

$$p_{ii}^{(k)} = p_{jj}^{(k)} = \frac{b_{k-1,\ i}^{(k-1)}}{\sqrt{(b_{i-1,\ i}^{(k-1)})^2 + (b_{k-1,\ j}^{(k-1)})^2}}$$

$$p_{ij}^{(k)} = -p_{ji}^{(k)} = \frac{-b_{i-1,\ j}^{(k-1)}}{\sqrt{(b_{i-1,\ k}^{(k-1)})^2 + (b_{i-1,\ j}^{(k-1)})^2}} \tag{5}$$

$$p_{rs}^{(k)} = \delta_{rs} \qquad \text{for other } (r,s).$$

Let the matrices $\{B_k\}$ and $\{P_k\}$ be defined by (1) and (5) for $k = 1, 2, \ldots, M$. Then the k elements of B_k whose indices correspond to the first k pairs listed in (4) are zero. B_M is in lower Hessenberg form.

OPERATION COUNT

Aside from the calculation of the nontrivial elements of P_k, the calculation of the nonzero elements of B_k in (1) involve 8n multiplications and 4n additions. Since this process must be performed M times ($M = \frac{(n-2)(n-1)}{2}$), the technique requires 8nM multiplications and 4nM additions to reduce a general matrix A to lower Hessenberg form.

These operation counts, however, are maximum limits on the number of additions and multiplications involved in the Givens method. To get an exact number for the operation count, we must consider the elements that are annihilated as we proceed through the algorithm, and therefore are not involved in

future computations. The number of times this occurs during the procedure is given by T, where

$$T = \sum_{k=0}^{n-3} [(n-2) - k](2K + 1) \text{ for } n \geq 3. \qquad (6)$$

T can be rewritten in terms of n only; that is,

$$T = \frac{1}{3}n^3 - \frac{3}{2}n^2 - \frac{77}{6}n + 12 \qquad (7)$$

Since we added in two multiplications and one addition each time the above situation occurred, we have that

$8nM - 2T$ multiplications,

and $4nM - T$ additions

are required to reduce A to lower Hessenberg form. Thus, the procedure requires $\frac{10}{3}n^3$ + (lower order terms) multiplications and $\frac{5}{3}n^3$ + (lower order terms) multiplications to affect the desired reduction.

In the event that A is symmetric, the procedure requires $\frac{4}{3}n^3$ + (lower order terms) multiplications to achieve the desired form. Hessenberg form in this instance is tridiagonal form. That is, A is a symmetric tridiagonal matrix. (See [2] for a description of the Givens' method.)

A1.2.2 HYMAN METHOD

A convenient technique for evaluating the characteristic polynomial $p_A(\lambda)$, when A is in lower Hessenberg form, makes use of the following theorem.

Theorem (Hyman): Let A be in lower Hessenberg form. If $a_{i,\ i+1} \neq 0$, $i = 1, 2..., n-1$, then we define the sequence of polynomials $m_i(\lambda)$ in the following fashion:

$$m_0 \equiv 1$$

$$m_i = -[a_{i1}m_0 + a_{i2}m_1 + \cdots + a_{i,\ i-1}m_{i-2} + (a_{ii}-\lambda)m_{i-1}] | a_{i,i+1}, \tag{8}$$

for $i = 1, 2, \cdots, n-1$. Then

$$p_A(\lambda) = \det(A - \lambda I) = (-1)^{n+1} a_{12}a_{23} \cdots a_{n-1,\ n} p(\lambda) \tag{9a}$$

where

$$p(\lambda) = a_{n1}m_0 + a_{n2}m_1 + \cdots + a_{n,\ n-1}m_{n-2} + (a_{nn}-\lambda)m_{n-1}. \tag{9b}$$

(See [2] for a description of the Hyman method.)

To evaluate the roots of $p(\lambda)$ by means of the standard iterative techniques, we should be able to calculate $p'(\lambda)$. This is done as follows:

$$\begin{aligned} m_0' &= 0 \\ m_1' &= 1 | a_{12} \end{aligned} \tag{10}$$

$$m_i'(\lambda) \quad \frac{-[a_{i1}m_0' + a_{i2}m_1' + \cdots + a_{i,i-1}m_{i-2}' + (a_{ii}-\lambda)m_{i-1}' - m_{i-1}]}{a_{i,i+1}}$$

$$p'(\lambda) = a_{n1}m_0' + a_{n2}m_1' + \cdots + a_{n,\ n-1}m_{n-2}' + (a_{nn}-\lambda)m_{n-1}' - m_{n-1}.$$

OPERATION COUNT FOR HYMAN'S METHOD

The method requires

$$ML = \sum_{i=1}^{n-1} i = \frac{1}{2}(n - 1)(n) = \frac{1}{2}[n^2 - n] \tag{11}$$

multiplications,

$$D = n - 1 \text{ divisions, and}$$

$$AD = \sum_{i=1}^{n} i = \frac{1}{2}n(n + 1) = \frac{1}{2}[n^2 + n]$$

additions to compute $p(\lambda)$. For $p'(\lambda)$, the method requires the same number of operations in all three cases.

A1.2.3 NEWTON-RAPHSON ITERATION METHOD

Choose x_0, and determine the sequence $\{x_n\}$ from the recurrence relation

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}, \quad n = 0, 1, 2, \ldots \tag{12}$$

Suppose we let

$$f(x) = x - \frac{F(x)}{F'(x)}, \tag{13}$$

and assume that F is twice continuously differentiable on the interval $I = [a, b]$. Let $F'(x) \neq 0$ for $x \in I$, and let the equation

$$F(x) = 0$$

have the solution $x = s$, where $x \varepsilon (a, b)$ (the open interval). Then, $f(s) = s$, and

$$f'(s) = 1 - \frac{[F'(s)]^2 - F(s)F''(s)}{[F'(s)]^2} = 1 - 1 = 0,$$

because $F(s) = 0$.

Now let $d_{n+1} = x_{n+1} - s$. Thus, by Taylor's remainder theorem,

$$\begin{aligned} d_{n+1} &= x_{n+1} - s \\ &= f(x_n) - s \\ &= f(x_n) - f(s) \\ &= f'(s)d_n + \tfrac{1}{2}f''(s + \theta_n d_n)d_n^2, \end{aligned}$$

where θ_n is an undetermined number between zero and one. But $f'(s) = 0$. Therefore,

$$d_{n+1} = \tfrac{1}{2}f''(s + \theta_n d_n)d_n^2. \qquad (14)$$

We conclude from (14) that the error at the (n+1)st step is proportional to the square of the error at the nth step. When Newton's method converges, it therefore converges quadratically. (See [1] for a description of the Newton-Raphson method.)

IMPLEMENTATION OF NEWTON'S METHOD

To use Newton's method to find the eigenvalues of the matrix A, we make a guess, say x_0, as to where a particular eigenvalue is located. Here is a theorem, due to Gerschgorin, which will help us make that guess.

Theorem (Gerschgorin): Let $A \equiv (a_{ij})$. We define absolute row and column sums by

$$r_i \equiv \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \qquad c_j \equiv \sum_{\substack{i=1 \\ i \neq j}}^{n} |a_{ij}| \tag{15}$$

Then,

(a) each eigenvalue lies in the union of the row circles R_i, $i = 1, 2, \ldots, n$, where

$$R_i \equiv \left\{ z : |z - a_{ii}| \leq r_i \right\}; \tag{16}$$

(b) each eigenvalue lies in the union of the column circles C_j, $j = 1, 2, \ldots, n$, where

$$C_j \equiv \left\{ z : |a - a_{jj}| \leq c_j \right\}; \tag{17}$$

(c) each component (maximal connected union of circles) of $\cup R_i$ or $\cup C_j$ contains exactly as many eigenvalues as circles. (Multiplicities are considered in the calculation.) (This theorem was lifted from [2; ch. 4].)

In the situation where the off-diagonal elements are small compared to the diagonal elements, the above theorem says to pick x_0 close to the diagonal element of a given row. When A is in lower Hessenberg form, there is only one non-zero off-diagonal element in the first row, two in the second, etc. Thus, some eigenvalues should be easily approximated by the values a_{ii} for small values of i. Obviously, trying to find the eigenvalues contained in the large circles would be far more difficult.

The following theorem will assist us further in attempting to select an x_0 that will assure convergence of the Newton-Raphson method.

Theorem: Let $F(x)$ be a real function, $F(x_0)/F'(x_0) \neq 0$, and let $h_0 = -F(x_0)/F'(x_0)$, $x_1 = x_0 + h_0$. Consider the interval J_0: $[x_0, x_0 + 2h_0]$ and assume that $F''(x)$ exists in J_0, that $\max_{J_0} |F''(x)| = M$ and

$$2|h_0|M \leq |F'(X_0)|. \tag{18}$$

Starting with x_0, we define the sequence $\{x_n\}$ by formula (12).

Then all x_n lie in J_0 and we have

$$x_n \longrightarrow s$$

where s is the only zero in J_0; s is a simple zero unless $s = X_0 + 2h_0$. Moreover, we have that

$$|s - x_{n+1}| \leq \frac{M}{2F'(x_n)} |x_n - x_{n-1}|^2. \tag{19}$$

(This theorem can be found in [3].)

Inequality (19) shows that the convergence is rapid when the appropriate J_0 is found.

In the event that s is a root of multiplicity p, we replace the Newton-Raphson formula by

$$x_{n+1} = x_n - p\frac{F(x_n)}{F'(x_n)}. \tag{20}$$

With this modification, the sequence $\{x_n\}$ not only converges, but it does so quadratically. The big problem here, however, is that p is not easily determined.

Let us now formulate the following theorem on multiplicities of size p. (See [3].)

Theorem: Let $F(x)$ have a root s of multiplicity p, and assume that $F^{(p+1)}(x)$ is continuous in a neighborhood of s. If we compute the sequence $\{x_n\}$ by means of (20) and if x_1 is sufficiently close to s, then all x_n exist and x_n converges to s. Moreover,

$$\frac{s - x_{n+1}}{(s - x_n)^2} \longrightarrow \frac{F^{(p+1)}(s)}{p(p + 1)F^{(p)}(s)}$$

as $n \longrightarrow \infty$.

A1.3 EVALUATION OF EIGENVECTORS

A1.3.1 GAUSSIAN ELIMINATION

Once the eigenvalue λ has been found, we must solve the equation

$$(A - \lambda I)X = 0, \tag{21}$$

where A is either the original matrix or the transformed matrix in lower Hessenberg form.

We now consider the method of Gassian elimination to solve equation (21). The object of this method is to transform the given matrix, $(A - \lambda I)$, into triangular form so that the solution is easily obtainable.

Quite obviously, it would be easier to transform $A - \lambda I$ to a triangular system if A were in Hessenberg form than if it were in unreduced form. Therefore, we select A to be in lower Hessenberg form. However, we must be aware of the fact that once an eigenvector is obtained, it must be transformed back to the original coordinate system via the formula

$$Y = PX,$$

where X is the computed eigenvector, and $P = \prod_{k=1}^{M} P_k$. (See eq. [1].) Hence, we must keep track of the transformation matrices used in equation (1).

To carry out the elimination process in this instance, we must modify the usual elimination procedure to annihilate the elements $a_{i,i+1}$ for $i = 1, 2, \cdots n-1$. We accomplish this by multiplying row $i + 1$ by $a_{i,\,i+1}/a_{i+1,\,i+1}$, subtracting row i from this multiple of row $i + 1$, and using this result as

the new row i. It should be clear that the new $a_{i, i+1}$ is zero. That is,.

$$a'_{i, i+1} = \frac{a_{i, i+1}}{a_{i+1, i+1}} a_{i+1, i+1} - a_{i, i+1} = 0.$$

In general,

$$a'_{i, k} = \frac{a_{i, i+1}}{a_{i+1, i+1}} a_{i+1, k} - a_{i, k}, \qquad (22)$$

where k = 1, 2, ··· i+1.

Once the lower triangular system $A' \equiv (a'_{ij})$ is obtained, we must compute a nontrivial solution of the equation

$$A'X = 0.$$

(A' is the triangularized form of $A - \lambda I$.)
Such a solution exists and can be computed by back-substitution if we assume that λ is an exact eigenvalue of A. We shall describe a more practical approach to this problem shortly.

OPERATION COUNT

To triangularize $A - \lambda I$, the count of the number of operations will be considered for two problems: the homogeneous problem,

$$A'X = 0,$$

and the non-homogeneous problem,

$$A'X = Y \quad (Y \neq 0).$$

For the number of multiplications, we have

$$\sum_{k=3}^{n} k + n = \frac{1}{2}n^2 + \frac{3}{2}n - 3,$$

for the homogeneous problem, and

$$\sum_{k=3}^{n+1} k + n+1 = \frac{1}{2}n^2 + \frac{5}{2}n - 4,$$

for the non-homogeneous problem.

For the additions there are

$$\sum_{k=2}^{n} k = \frac{1}{2}n(n + 1) - 1 = \frac{1}{2}n^2 + \frac{1}{2}n - 1,$$

for the homogeneous case, and

$$\sum_{k=3}^{n+1} k = \frac{1}{2}n^2 + \frac{3}{2}n + 1 - 3 = \frac{1}{2}n^2 + \frac{3}{2}n - 2,$$

for the non-homogeneous case.

For the number of divisions, we have n - 1 operations for both the homogeneous and non-homogeneous problems.

A1.3.2 INVERSE ITERATION

What is probably a more practical approach to the problem of solving for eigenvectors is known as the method of inverse iteration or Wielandt iteration. (See [4; p. 142].) The procedure is simply described in the following fashion: we

form the sequence of vectors $\{X_n\}$ by the relation

$$(A - \lambda I)X_{n+1} = X_n \quad [X_n = (x_n^{(1)}, \ldots, x_n^{(n)})],$$

where λ is an approximate eigenvalue of A and X_1 is an arbitrary normalized starting vector. If A is assumed to be in lower Hessenberg form, then Gaussian elimination, as described above, is first applied to the system $A - \lambda I$ before the iteration process begins. Once $A - \lambda I$ has been triangularized, the above equation is solved by means of back-substitution, provided we know the vector X_n. This substitution process involves taking the values $x_{n+1}^{(1)}, \ldots, x_{n+1}^{(i-1)}$ and plugging them into the ith equation to solve for $x_{n+1}^{(i)}$. (See note below.) (Actually, back-substitution in this case really refers to forward-substitution because the system $A - \lambda I$ is in lower triangular form.)

The operation count for this substitution process is as follows:

$$\sum_{k=1}^{n-1} k = \frac{1}{2}(n-1)n = \frac{1}{2}n^2 - \frac{1}{2}n \text{ multiplications,}$$

$$\sum_{k=1}^{n-1} k = \frac{1}{2}n^2 - \frac{1}{2}n \text{ additions, and}$$

n divisions.

NOTE: The triangularization process on $A - \lambda I$ is performed once, but those same elementary operations must be performed on X_n for each n.

A1.3.3 SYMMETRIC MATRICES

The operation counts for the case where A is symmetric and the problem is non-homogeneous are as follows:

for Gaussian elimination, we have

$6 + (n-2)4$	multiplications,
$3(n-1)$	additions, and
$n-1$	divisions;

for forward or back-substitution, we have

$n-1$	multiplications,
$n-1$	additions, and
n	divisions.

The number of operations in this instance is greatly reduced over the number required in the non-symmetric case because A is in tridiagonal form in addition to being symmetric.

A1.4 A PRIORI ERROR ESTIMATES FOR THE GIVENS' METHOD

We begin by restating equation (1)

$$\begin{aligned} B_0 &\equiv A, \\ B_{k+1} &\equiv P_k^* B_k P_k, \qquad 0 \le k \le M-1. \end{aligned} \tag{23}$$

It is important to note that P_k is unitary. Hence

$$P_k^* P_k = I,$$

and

$$||P_k|| = 1,$$

where $||\cdot||$ is the spectral norm for matrices.

NOTE: The spectral norm for any matrix A is given by

$$||A|| = \sup ||AX|| \quad \text{(taken over all X with } ||X||=1\text{)},$$

where $||X|| = (x_1^2 + x_2^2 + \cdots + x_n^2)^{\frac{1}{2}} \quad (X = (x_1, x_2, \cdots, x_n))$.

The error bounds that we are interested in center around the computation of P_k and B_{k+1}. We let B_k be the actual kth computed matrix in the procedure, and P_k the exact kth unitary matrix, determined by the process, corresponding to B_k.

In the computation of P_k, rounding errors are committed and we actually compute a matrix which we shall denote by $\overline{P_k}$. Therefore, we get an error matrix S_k which is defined by the relation

$$\overline{P_k} = P_k + S_k. \tag{24}$$

The computation of B_{k+1} involves further rounding errors. In this regard, we define F_k by the equation

$$B_{k+1} = P_k^* B_k P_k + F_k. \tag{25}$$

Thus, the matrix F_k represents the difference between the computed B_{k+1} and the exact transform of the computed B_k; that is, $P_k^* B_k P_k$. Our main objective will be to compute error bounds for S_k and F_k. (For the details of the a priori error estimates for the Givens' method, see [5].)

When we combine the system of equations defined by (25), we have

$$B_{k+1} = Q_{k0}^* B_0 Q_{k0} + Q_{k1}^* F_0 Q_{k1} + \cdots Q_{kk}^* F_{k-1} Q_{kk} + F_k$$

where (26)

$$Q_{ki} = P_i P_{i+1} \cdots P_k.$$

Q_{ki} is exactly orthogonal (unitary) by equation (26).

If we define G_k by the relation

$$G_k = Q_{k1}^* F_0 Q_{k1} + \cdots + Q_{kk}^* F_{k-1} Q_{kk} + F_k,$$

then using the fact that $||Q_{ki}|| = 1$

$$||G_k|| \le ||F_0|| + \cdots + ||F_k||. \tag{27}$$

We can now rewrite (26) as

$$B_{k+1} = Q^*_{k0} B_0 Q_{k0} + G_k. \tag{28}$$

From (28), we can conclude that

$$||B_0|| - ||G_k|| \leq ||B_{k+1}|| \leq ||B_0|| + ||G_k||. \tag{29}$$

We assume that B_0 is normalized so that

$$||B_0|| \leq N - \delta$$

and that a bound on $||G_{M-1}||$, based on this assumption, can be found. Then by induction and (29),

$$N - 2\delta \leq ||B_{k+1}|| \leq N \qquad (0 \leq k \leq M-1). \tag{30}$$

(It is assumed that $\delta < N$.) Thus, under these circumstances, a bound can be placed on the variation of $||B_k||$ due to rounding errors.

In [5], the above analysis was used to compute bounds on $||F_k||$ and ultimately $||G_k||$ with the assumptions that t binary digits makeup a word, fixed-point arithmetic is used, and inner products are accumulated. Also, the error is given for real computation only.

In addition to these assumptions, we assume that B_0 is normalized so that $||B_k|| \leq 1$ for all k. The scaling (normalization), however, is performed when the analysis has been completed.

We first consider a typical stage in the process where the first i-1 rows have been reduced to lower Hessenberg form, and positions i+2, i+3, ..., j-1 in row i have been annihilated. The next step in the process concerns the annihilation of the element i, j. Furthermore, $(a^2_{i,\ i+1} + a^2_{ij})$ is accumulated exactly and then the integer k is determined so that

$$\frac{1}{4} \leq 2^{2k}(a^2_{i,\ i+1} + a^2_{ij}) < 1. \tag{31}$$

(Note that the statement of the Givens' theorem has us annihilating element i-1, j instead of i, j so the indexing is slightly different here.)

To avoid the details of the analysis involved in estimating the round-off error, we simply state the results given in [5] for non-symmetric matrices.

With the scaling shown in (31), we arrive at the following results:

$$||S_k|| \leq \frac{\frac{1}{2}2^{-t}}{\frac{1}{2}(1-2^{-t})} + \sqrt{\frac{1}{2}}\,2^{-t} \tag{32}$$

$$< (1.71)2^{-t};$$

$$||F_k|| \leq [2(1.71)2^{-t} + (1.71)^2 2^{-2t}]$$

$$+ \left[1.21 + \sqrt{\frac{1}{2}(n-i+1)} + \sqrt{\frac{1}{2}i}\right]2^{-t} + 2^{-t}$$

$$\leq \left[5.7 + \sqrt{\frac{1}{2}(n-i+1)} + \sqrt{\frac{1}{2}i}\right]2^{-t}, \tag{33}$$

and summing over k (see inequality [27])

$$||G_k|| \le ||G_{M-1}|| \le [2.9n^2 + \frac{\sqrt{2}}{5}n^{5/2} + \frac{2\sqrt{2}}{15}n^{\frac{5}{2}}]2^{-t}$$

$$= [2.9n^2 + \frac{\sqrt{2}}{3}n^{5/2}]2^{-t}. \qquad (34)$$

(Reminder: n is the size of the matrix.)

We now let δ equal the right side of (34), and normalize B_0 so that

$$||B_0|| \le 1-\delta.$$

Then $||B_k||$ is contained in the manner given in (30). We thus have our a priori estimations on the variation of $||B_k||$.

A1.5 WELL-POSEDNESS AND A POSTERIORI ERROR ESTIMATES

We now present a sequence of results which can be found in [2, ch 4]. We shall assume throughout that A is diagonalizable; that is, A possesses a basis of eigenvectors. (A is diagonalizable if and only if it has a basis of eigenvectors.)

The Gerschgorin circle theorem, stated earlier, plays an important role in proving the following theorem on well-posedness.

Theorem: Let A be of order n and have n linearly independent eigenvectors. For any fixed matrix C, with $||C|| = ||A||$, we define the perturbed matrix

$$A(\varepsilon) \equiv A + \varepsilon C.$$

Then if λ is any eigenvalue of A, there is an eigenvalue $\lambda(\varepsilon)$ of $A(\varepsilon)$ such that

$$|\lambda(\varepsilon) - \lambda| \leq K|\varepsilon| \qquad (K = \text{constant}) \tag{35}$$

for all small ε. Moreover, if λ is simple (multiplicity = 1)

$$\lim_{|\varepsilon| \to 0} \frac{\lambda(\varepsilon)-\lambda}{\varepsilon} = \frac{Y^*CX}{Y^*X},$$

where X and Y are, respectively, left and right eigenvectors of A corresponding to λ. That is, X satisfies the equation $AX = \lambda X$ and Y satisfies the equation $Y^*A = \lambda Y^*$.

If we let P be the matrix whose columns are the right eigenvectors of A, then in inequality (35)

$$K = ||C|| \cdot ||P^{-1}|| \cdot ||P||. \tag{36}$$

Inequality (35) with K defined by means of (36) was derived by Bauer and Fike (see [2; p. 139]).

When A is normal ($A^*A = AA^*$), then it can be assumed that P is unitary; thus $||P^{-1}|| = ||P|| = 1$ and, by (35) and (36), the problem is well-posed for all eigenvalues.

A1.6 A RESULT ON A POSTERIORI ERROR ESTIMATES

The a priori error estimates provided earlier usually cannot be relied upon for sharpness. However, once the eigenvalues and eigenvectors have been computed, a posteriori error estimates, using these computed vectors and scalars, can be obtained. Very often, error estimates of this type are a lot less crude than those of the a priori type. We now present a theorem which provides a useful way of obtaining error estimates based on computed eigenvalues and eigenvectors.

Theorem: Let A be of order n, and have a set of n linearly independent eigenvectors $\{U_i\}$ with $\{\lambda_i\}$ as the corresponding eigenvalues. Also let $U \equiv (U_1, U_2, \ldots, U_n)$. If for some $\varepsilon > 0$,

$$||AX - \lambda X|| \leq \varepsilon ||AX||,$$

then

$$\min_{\lambda_j \neq 0} \left| 1 - \frac{\lambda}{\lambda_j} \right| \leq \varepsilon ||U|| \cdot ||U^{-1}||.$$

If $||AX - \lambda X|| \leq \varepsilon||X||$,

then $\min_i|\lambda - \lambda_i| \leq \varepsilon||U^{-1}|| \cdot ||U||$.

Furthermore, when A is normal

$$\min_i|\lambda - \lambda_i| \leq \frac{||N||}{||X||},$$

where $N = AX - \lambda X$.

A1.7 AN ALGORITHM FOR SPARCE MATRICES

A1.7.1 INTRODUCTION

We now describe and consider an approach to the eigenvalue problem which takes sparcity into account. This approach takes as its basis the method of Gaussian elimination to reduce the matrix to Hessenberg form. Other important parts to this algorithm include Laguerre's iterative scheme and Hyman's method (mentioned previously), which combine to compute the eigenvalues. When the eigenvalues have been obtained, we resort to the aforementioned inverse initiation technique to calculate the eigenvectors. (Most of the ideas in what follows can be attributed to T. Papathomas and O. Wing [7], [8] and [9].)

The Given's technique, which was described earlier, is not considered here because it is not conducive to sparce matrix computations. In the case of sparce matrices, we are concerned not only with stability of computations but with the generation of "fill-ins"--the creation of nonzero elements where zeros have previously existed. Experience has shown that the Givens' method produces a rather large number of fill-ins. Other well-known techniques, such as the Householder and Jacobi methods and the QR transformation, also produce unacceptable numbers of fill-ins. One of the few methods that can be implemented to yield a relatively small number of fill-ins is that of Gaussian similarity transformations. (See [7], [8] or [9].) We describe this technique after the next section.

A1.7.2 REDUCING THE SIZE OF THE MATRIX

It is sometimes possible to determine some of the eigenvalues before performing any actual computations. That is, it is conceivable that a permutation matrix P exists such that $A' = P^{-1}AP$ has the form

$$A' = \begin{bmatrix} A'_{11} & A'_{12} & A'_{13} \\ 0 & A'_{22} & A'_{23} \\ 0 & 0 & A'_{33} \end{bmatrix},$$

where the submatrices A'_{11} and A'_{33} are upper triangular. Clearly, if A can be transformed to the above form, then it is desirable to do so because the diagonal elements of A'_{11} and A'_{33} are eigenvalues of A. Therefore, once A has been transformed, it is necessary to find the eigenvalues of A'_{22} only, and so we shall henceforth view A as the smaller and more manageable matrix A'_{22}.

In practice, we perform simultaneous row and column interchanges to transform A to A'. First we look for a column that has all zeros below its topmost element. Then we interchange this column with the first column, while the corresponding row interchanges are concurrently performed. Next we look for a column that has all zeros below the element, which is second from the top. If such a column exists, we interchange it with the second column and then perform the necessary row interchanges. This process continues until we can no longer find a column with zeros below its kth element. This process isolates A'_{11} for us. In the kth step of the process which isolates A'_{33}, we take row r_k, which has zeros to the left of its $n - k + 1$th element, and interchange it with row $n - k + 1$. Thus, A has been transformed to A'.

Having performed the above transformation, it is advisable to apply a scaling procedure which reduces the size of $||A||$. This stage of our method is included to improve numerical stability. We do not include this algorithm here, and so the reader is referred to [11] where this procedure, due to Parlett and Reinsch, can be found.

A1.7.3 REDUCTION TO HESSENBERG FORM

If we assume that $A^{(k-1)}$ is the matrix obtained after the first k-1 columns have been reduced, then we perform the kth step to get $A^{(k)}$ as follows: to reduce all the elements in column k below row k+1 to zero, we must first select a pivot element. If the pivot selection is not $a_{k+1,k}^{(k-1)}$, then an interchange of rows (followed by an interchange of columns for the purpose of completing a similarity transformation) must be effected. If we denote the resultant matrix by $\hat{A}^{(k-1)}$, then we reduce the elements in the kth column by multiplying the pivotal row by $\mu_i^{(k)} = \hat{a}_{ik}^{(k-1)} / \hat{a}_{k+1,k}^{(k-1)}$ and add it to row i for $i \geq k+2$. To complete the similarity transformation, we multiply column i by $-\mu_i^{(k)}$ and add it to column k+1, where $i \geq k+2$.

Remark: The reduction of column k corresponds to a multiplication on the left (premultiplication) of $\hat{A}^{(k-1)}$ by a matrix M_k, and the operations to complete the similarity transformation correspond to a multiplication on the right (postmultiplication) of $M_k\hat{A}^{(k-1)}$ by the matrix M_k^{-1}. (This procedure can be found in [7].)

A1.7.4 THE TRANSFORMATION TO UPPER BANDED FORM

When the reduction process is performed, it is desirable that the matrix be in upper banded form. Before providing the definition of upper banded form, we state the definition of a corner point.

Definition: An element a_{ij} of a matrix A is a corner point if and only if

$$a_{ij} \neq 0,$$

$$a_{m\ell} = 0 \text{ for } m \leq i - 1,\ \ell \geq j,$$

and

$$a_{i\ell} = 0 \text{ for } \ell > j.$$

We assume, for instance, that a non-zero element in the upper righthand corner of the matrix vacuously satisfies the definition of a corner point.

Definition: A matrix is said to be in upper banded form if and only if it has more than one corner point. That is, a matrix is in upper banded form when it has the structure shown in Figure 1.

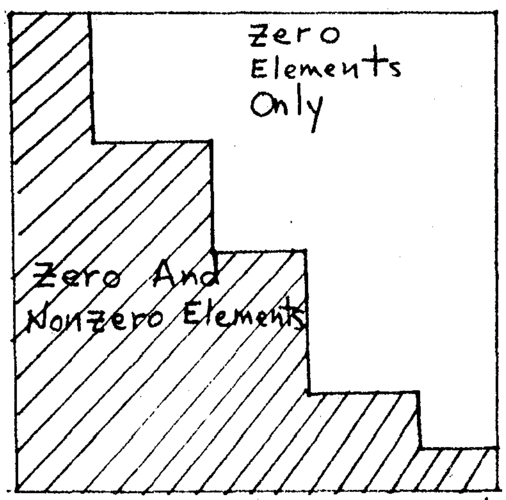


Figure 1

A careful examination of the reduction method reveals that when no row or column interchanges are effected (no pivoting) a pivot element changes only those elements below it and to the right. Therefore, when a matrix, which is in upper banded form, is reduced to Hessenberg form using Gaussian similarity transformations without pivoting, it remains in upper banded form. Furthermore, its corner elements are unchanged.

The above statement is saying that no fill-ins are produced in the white area of Figure 1 when Gaussian similarity transformations are used without pivoting. Thus, it is clear that we increase our chances of obtaining a sparce Hessenberg matrix when the original matrix has the structure shown in Figure 1. We shall be more specific about this in the section on sparce reduction when algorithms 2 and 3 are combined.

We now provide the algorithm, which is given in [9], to transform the matrix A into upper banded form.

Definition: The m-distance, denoted $d(i,j)$, of the position (i,j) in the matrix A is given by

$$d(i,j) = j - i.$$

The m-distance is a measure of the distance that a_{ij} is away from the main diagonal.

The algorithm that we are about to present performs row and column interchanges such that the number of zeros, say z, in the white area of Figure 1 is maximized. The approach taken here is to select a corner element a_{ij} and see if z is increased by moving this element into the shaded region. We look for a column $k < j$ such that

$$a_{\ell k} = 0 \text{ for } \ell \leq i. \tag{37}$$

It then appears as though z will increase if we interchange columns j and k. However, we must also interchange rows j and k, and when this takes place, new non-zero elements may be introduced into the white area. Thus, it is possible for z to actually decrease. Clearly, we need some criteria to assure us that z will increase when the above row and column interchanges are performed. So, we employ the following criteria where it is assumed that a_{jq} is the furthest non-zero element to the right in row j, D and δ are the maximum and minimum m-distances, respectively, associated with the current set of corner elements, and c is the number of current corner elements.

Criterion 1: Do not perform the transformation unless the m-distance of (j,q) does not exceed δ.

Criterion 2: Do not perform the transformation unless z increases as a result of the transformation.

Criterion 3: Replace δ with D in criterion 1.

Of the three criteria given above, criterion 1 is the most difficult to satisfy. However, when this criterion is satisfied, a significantly increased z usually results. Criterion 2 guarantees an increasing z while criterion 3 is the weakest and least reliable of the three.

To implement the above criteria, we take a corner element a_{ij} and search for a column $k < j$ such that (37) is satisfied. We now apply criterion 1. If it is satisfied, we perform the appropriate row and column interchanges. The new corner elements are then determined and we repeat the above procedure with one of the new corner elements. If for any column $k < j$

that satisfies (37), the corresponding (j,q) position does not satisfy criterion 1 or if no column $k < j$ which satisfies (37) can be found, then we move on to another corner element. When we are unable to perform the transformation for any corner element of the current corner element set, criterion 2 is then used. This process continues until all three criteria have been exhausted.

Remark: We remark that in the procedure just outlined, the corner elements are selected according to m-distances. That is, we start with the largest m-distance and work our way down to the smallest.

Also, we use criterion 2 as a backup to criteria 1 and 3 to assure that z ultimately increases. It is possible for z to decrease locally but increase globally when criterion 1 (or 3) is involved. The application of criterion 2 in these instances serves as a means of preventing a globally decreasing z and an endless looping process.

A1.7.5 SPARCE REDUCTION TO HESSENBERG FORM

We now discuss several algorithms which are used in conjunction with the previously described method of Gaussian similarity transformations. In the selection of any algorithm to reduce a matrix to Hessenberg (almost triangular) form, we must keep in mind two important notions: one is the number of fill-ins generated, and the other is numerical stability. Unfortunately, any preoccupation with one of these notions often results in a sacrifice of the other. Thus, a good algorithm should consist of a compromise of the two above notions.

As far as fill-ins are concerned, we consider the three regions, X, Y and Z shown in Figure 2. (Again we refer the reader to [7].) It is these regions of the matrix where the changes occur during the kth step. Region X contains fill-ins that may include the pivot element for the next step. Thus, these fill-ins

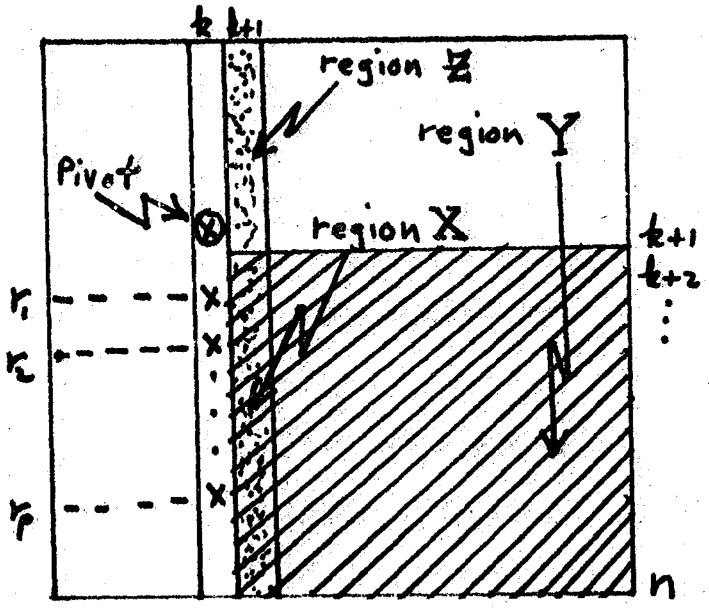


Figure 2

could be responsible for future fill-ins. However, none of these fill-ins will appear in the final Hessenberg matrix. Region Y contains fill-ins that could create future fill-in generation, and these may or may not appear in the Hessenberg matrix. Region Z contains fill-ins that will certainly not generate future ones, but these will definitely appear in the final Hessenberg matrix. If we denote by n_X, n_Y and n_Z, the number of fill-ins in regions X, Y and Z, respectively, during the kth step, then algorithm 1 can be stated as follows: we simply choose as our pivot that element in column k (below row k) which minimizes T, where

$$T = n_X + n_Y + n_Z.$$

This algorithm is due to Tewarson [12], and its main limitation is that it does not take computational stability into account.

Algorithm 2, which also does not consider computational stability, is formulated as follows: we simply take a matrix, which has been transformed to upper banded form as previously described, and reduce it to Hessenberg form via Gaussian similarity transformations without any reordering (interchanging) of rows or columns. As previously stated, the upper banded structure of the matrix is preserved when the reduction process is executed this way. Hence, as far as fill-in is concerned, the above algorithm is usually favorable. However, the amount of round-off error could be very prohibitive, and, moreover, we are not always guaranteed that the original matrix can be transformed to upper banded form. Later, we shall show how this algorithm could be combined with algorithm 3, which does take stability of computations into account. (Algorithm 2 is formulated in [7].)

In considering the problem of numerical stability, we could take as our pivot element that entry in column k (below row k) which is largest in absolute value. However, this approach, while usually providing stable computations, completely ignores the phenomenon of fill-in. In [7], an algorithm which takes both phenomena into account, is formulated. We present that formulation here as algorithm 3.

Algorithm 3 states that at the kth step we select as a pivot that element (in column k, below row k) which satisfies two conditions. First, it must minimize W, where

$$W = c_X n_X + c_Y n_Y + c_Z n_Z \tag{38}$$

and c_X, c_Y and C_Z are constants as yet undetermined. Second, it must satisfy the following stability condition:

Let $b_{k+1,k}$, $b_{k+2,k}$, ..., b_{nk} be the elements in column k under consideration and let

$$\beta_M = \max_{k+1 \le i \le n} |b_{ik}|$$

and

$$\beta_m = \min_{k+1 \le i \le n} |b_{ik}| \ ;$$

an element $b_{ik}(k+1 \le i \le n)$ is disqualified as a pivot candidate if

$$|b_{ik}| < \beta_m + \alpha(\beta_M - \beta_m), \ 0 \le \alpha \le 1. \qquad (39)$$

Obviously, the closer α is to one the more we ignore fill-in and the closer it is to zero the more we ignore computational stability.

Assuming that the original matrix can be transformed to upper banded form, we can combine algorithms 2 and 3. In this combined approach, we test the first non-zero element of the set $s = \left\{b_{k+1,k}, \ldots, b_{n,k}\right\}$ to see if it satisfies inequality (39). If it does not, then (without any row or column interchanges) we immediately reduce column k using that first non-zero member of s as one pivot element. If this entry does satisfy (39), then we resort to algorithm 3 without any regard for algorithm 2. The important idea in this resultant procedure is that an attempt is made to maintain the upper banded structure of the matrix as it is being reduced while still taking computational stability into account. It is clear that when implementing algorithms 2 and 3 this way we

may be selecting a pivot element which does not minimize W (see (38)). However, it is the fill-ins above the main diagonal that concern us most and it is these fill-ins that our new algorithm attempts to avoid. In this fashion, we tend to concentrate the non-zero elements toward the main diagonal during the reduction process.

At this point, it is probably worth explaining how n_X, n_Y and n_Z are determined. Clearly, fill-in is generated only when one row (or column) is added to another. In the row where the result of the sum will be written, we check for the locations containing zero elements. The number of non-zero elements in the corresponding locations of the other row is equal to the number of fill-ins generated by the addition. It is important to realize that no computations are needed to determine the fill-in count.

Having reduced the matrix A to what we hope is a sparce Hessenberg matrix H, we can proceed to finding the eigenvalues using the Laguerre method.

A1.7.6 LAGUERRE'S METHOD

As an alternative to the Newton-Raphson iteration method, we present the Laguerre method, which has the advantage of not requiring the deflation (reduction in size according to the multiplicity of λ_i) of the matrix H (the Hessenberg matrix from the reduction process) when an eigenvalue has been accepted (See [10] and [7].) For any polynomial p(x) of degree n with roots λ_1, λ_2, ..., λ_n (perhaps not all distinct), we define the following:

$$s_1(x) = \frac{p'(x)}{p(x)} = \sum_{i=1}^{n} \frac{1}{x - \lambda_i} , \qquad (40)$$

and

PAGE 79

IS MISSING

IN **ORIGINAL**

$$s_2(x) = \frac{(p'(x))^2 - p(x)p''(x)}{(p(x))^2} = \sum_{i=1}^{n} \frac{1}{(x - \lambda_i)^2} . \qquad (41)$$

The Laguerre scheme constructs a sequence $\{x_k\}$ that converges cubically to a simple root by means of the formula

$$x_{k+1} = x_k - \frac{n}{s_1(x_k) \pm \sqrt{(n-1)(ns_2(x_k)) - s_1^2(x_k)}} \qquad (42)$$

If the polynomial p(x) has real roots only, it is known that the real line can be divided into n_1 contiguous intervals (n_1 = number of distinct roots) such that for any starting value x_0 in a given interval the sequence x_k converges monotonically to the root included in that interval.

In showing that no deflation is needed once an eigenvalue has been found, we present the discussion given in [7] or [8]. Let $p_k(x)$ denote the function p(x) with k roots removed. Thus,

$$p_0(x) = p(x),$$

and

$$p_k(x) = \frac{p(x)}{\prod_{i=1}^{k}(x - \lambda_i)} .$$

In this situation, $s_1^{(k)}(x)$ and $s_2^{(k)}(x)$ are defined in (40) and (41), respectively, where p(x) and its derivatives are replaced by $p_k(x)$ and its derivatives. However, since $p_k(x)$

and its derivatives are not available, we evaluate $s_1^{(k)}(x)$ and $s_2^{(k)}$ as follows:

$$s_1^{(k)}(x) = s_1(x) - \sum_{i=1}^{k} \frac{1}{x - \lambda_1} ,$$

and

$$s_2^{(k)}(x) = s_2(x) - \sum_{i=1}^{k} \frac{1}{(x - \lambda_i)^2} .$$

A1.7.7 STARTING AN ITERATION

Finding a proper starting point in the Laguerre iteration process is extremely important. We do this by making use of the fact that Laguerre iterations are invariant under all Möbuis transformations

$$Tz = \frac{az + b}{cz + d}, \ a, b, c, d \text{ real}, \begin{vmatrix} a & b \\ c & d \end{vmatrix} \neq 0.$$

To illustrate this, we take the special case where $Tz = \frac{1}{z}$. In this connection, we consider the polynomials $p(x)$ and $\frac{1}{p(x)}$. The above statement in this context says that if z' is obtained from z using (42) for the polynomial $\frac{1}{p(x)}$, then $\frac{1}{z'}$ is obtained from $\frac{1}{z}$ using (42) for the polynomial $p(x)$.

We use the above ideas to select a starting value which has a greater magnitude then the eigenvalue with greatest absolute value. Let us consider the polynomial which is reciprocal to the polynomial, det $(H - xI)$. If z' is obtained from (42) using the reciprocal polynomial with zero replacing x_k, then, by the above comments, $\frac{1}{z'}$ is the Laguerre iterate corresponding to the point at $x = \infty$. (Note: Under the Möbuis transformation, $Tz = \frac{1}{z}$, the point at $x = 0$ gets assigned to the point at

$x = \infty$ on the extended number line.) We define $\sigma_1(x)$ and $\sigma_2(x)$, respectively, in terms of the reciprocal polynomial just as $s_1(x)$ and $s_2(x)$ are defined in terms of the polynomial det $(H - xI)$.

Thus, assuming that $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of H,

$$\sigma_1(0) = \sum_{i=1}^{n} \frac{1}{0 - \frac{1}{\lambda_i}} = -\sum_{i=1}^{n} \lambda_i,$$

and

$$\sigma_2(0) = \sum_{i=1}^{n} \frac{1}{(0 - \frac{1}{\lambda_i})^2} = \sum_{i=1}^{n} \lambda_i^2$$

Since the trace of a matrix is unchanged by a similarity transformation,

$$\sum_{i=1}^{n} \lambda_i = \mathrm{tr}(H) = \sum_{i=1}^{n} h_{ii},$$

and

$$\sum_{i=1}^{n} \lambda_i^2 = \mathrm{tr}(H^2) = \sum_{i=1}^{n} h_{ii}^2 + 2 \sum_{i=1}^{n-1} h_{i,i+1}n_{i+1,i}.$$

Since we can now compute $\sigma_1(0)$ and $\sigma_2(0)$, we get, using (42), that

$$\frac{1}{z'} = -\frac{1}{n}[\sigma_1(0) + \sqrt{(n-1)[n\sigma_2(0) - (\sigma_1(0))^2]} .$$

We take $\frac{1}{z}$, as our first starting value when λ_i are real for all i. Once the first eigenvalue has been accepted, we use a Newton iteration with this newly found eigenvalue to select a starting value for the next eigenvalue. (The technique just discussed as well as the procedure for finding subsequent starting values can be found in [10].)

A1.7.8 THE COMPUTATION OF $p(\lambda)$, $p'(\lambda)$ AND $p''(\lambda)$

A relatively efficient means of computing $p(\lambda)$, $p'(\lambda)$ and $p''(\lambda)$ is very important in carrying out the Laguerre iteration technique. The reduction process described earlier was formulated with this problem in mind, and so we manipulate (8), (9) and (10) to emphasize why reordering is used to arrive at a sparce Hessenberg matrix H. Since the matrix A in the Hyman theorem given earlier is in lower Hessenberg form, we assume that H is also in that form. However, the a_{ij}'s (equation (8), (9) and (10)) become h_{ij} in the equations that follow.

We must be aware of the fact that the Hyman method involves taking multiples of columns 2, ..., n of $H - \lambda I$ and adding each to column 1 so as to annihilate the first $n - 1$ elements of that column. This establishes the m_i's in equations (8) and (9b); that is, m_1 is the multiple for column 2, m_2 is the multiple for column 3, etc. The element appearing in the lower lefthand corner of the new matrix is the $p(\lambda)$ of equation (9a), whose justification follows from the fact that the determinant of the new matrix is the same as $\det(A - \lambda I) = \det(H - \lambda I)$.

The constant multiple of $p(\lambda)$ in (9a) cancels out in the definitions of $s_1(x)$ and $s_2(x)$ in (40) and (41), respectively,

and so we simply ignore $P_A(\lambda)$. Remembering that H is in lower Hessenberg form, we let

$$H - \lambda I = [W_1 \vdots W_2 \cdots \vdots W_n].$$

Equations (8) and (9a) can be reformulated as follows:

$$W_1 + m_1 W_2 + \cdots + m_{n-1} W_n = e_n p(\lambda), \tag{43}$$

where $e_n = [0, 0, \ldots, 1]^t$. We can rewrite (43), so as to show explicitly what is involved in the computation of the m_i followed by the calculation of $p(\lambda)$, to get that

$$WM = C_0 \tag{44}$$

$$p(\lambda) = M^t U_0 + h_{n1} \tag{45}$$

where W is the matrix $H - \lambda I$ with its first column and last row removed,

$$M^t = [m_1, m_2, \ldots, m_{n-1}],$$

$$C_0^t = [-h_{11} + \lambda, -h_{21}, \ldots, -h_{n-1,\ 1}],$$

and

$$U_0^t = [h_{n2}, h_{n3}, \ldots, h_{n,n}].$$

It is clear that W is an n-1 by n-1 lower triangular matrix and, therefore, the solution to (44), which corresponds to (8), is computed in a straightforward manner. To get $p'(\lambda)$, we differentiate (44) and (45) yielding

$$WM' = C_1, \tag{46}$$

and

$$p'(\lambda) = (M')^t U_0 - m_{n-1}, \tag{47}$$

where $C_1^t = [1, m_1, m_2, \ldots, m_{n-2}]$; if we differentiate (46) and (47) we get that

$$WM'' = C_2,$$

and

$$p''(\lambda) = (M'')^t U_0 - 2m'_{n-1},$$

where $C_2^t = [0, 2m'_1, 2m'_2, \ldots, 2m'_{n-2}]$.

It is very important to observe that solutions to $WM^{(i)} = C_i$ $(i = 0, 1, 2)$ are repeatedly sought in order to compute $p(\lambda)$, $p'(\lambda)$ and $p''(\lambda)$. As a result, it is essential to obtain a sparce W to facilitate computations, and thus, the emphasis on reordering during the reduction process is justified. (The above analysis can be found in [7], where the matrix H is assumed to be in upper Hessenberg form.)

Now that we have shown how the eigenvalues are found by means of the Laguerre iteration procedure, the reader is referred to section A1.3.2 on the method of inverse iteration to find the corresponding eigenvectors.

A1.7.9 ERROR ANALYSIS OF GAUSSIAN REDUCTION

For completeness, we attempt to provide a priori error analysis of Gaussian reduction in this section. Out of convenience, however, we shall consider a different formulation

of this reduction technique, although the error bounds will be similar to those associated with the process used in this report.

We know that the Hessenberg matrix H satisfies the relation

$$N^{-1}AN = H, \tag{48}$$

where A is the original matrix. If we rewrite (48) as

$$AN = NH, \tag{49}$$

then by equating the corresponding columns on both sides of (49), we arrive at the recursive relations

$$h_{ir} = a_{ir} + \sum_{k=r+1}^{n} a_{ik}n_{kr} - \sum_{k=1}^{i-1} n_{ik}h_{kr} \quad (k = 1, \ldots, r+1), \tag{50}$$

and

$$n_{i,r+1} = \left(a_{ir} + \sum_{k=r+1}^{n} a_{ik}n_{kr} - \sum_{k=1}^{r} n_{ik}h_{kr}\right) \Big/ h_{r+1,r} \tag{51}$$

$(i = r+2, \ldots, n)$. h_{11} and h_{21} can be found on the first pass, because it is well known that the first column of N is $(1, 0, 0, \ldots, 0)^t$. Then h_{21} is used to compute n_{32} through n_{n2}. h_{12}, h_{22} and h_{32} are then computed with these new values of n_{rs} and the process continues until the nth column of H has been computed.

To get an idea as to the errors involved in computing H,

we assume that fixed-point arithmetic with inner product accumulation is used. That is, the inner products of (50) and (51) are accumulated exactly and then rounding occurs when the final number is stored in memory to t significant places. Hence, from (50) and (51)

$$h_{ir} = a_{ir} + a_{i,r+1}n_{r+1,r} + \cdots + a_{in}n_{nr} - n_{i1}h_{1r} - \cdots - n_{i,i-1}h_{i-1,r} + E_{ir}, \tag{52}$$

where $|E_{ir}| \leq \frac{1}{2}(2^{-t})$ $(i \leq r+1)$,

and

$$n_{i,r+1} = \left(a_{ir} + a_{i,r+1}n_{r+1,r} + \cdots + a_{in}n_{nr} - n_{i1}h_{1r} - \cdots - n_{ir}h_{rr}\right) / h_{r+1,r} + N_{ir}, \tag{53}$$

where $|N_{ir}| \leq \frac{1}{2}(2^{-t})$ $(i > r+1)$. If we multiply both sides of (53) by $h_{r+1,r}$, then $n_{i,r+1}h_{r+1,r}$ is computed to within an error of E_{ir} $(i > r+1)$, where $E_{ir} = h_{r+1,r}N_{ir}$. Now we let

$$(f_{ir}) = F = AN - NH, \tag{54}$$

and the above equations allow us to conclude that

$$|f_{ir}| \leq \begin{cases} \frac{1}{2}(2^{-t}), & \text{for } i \leq r+1 \\ \frac{1}{2}h_{r+1,r}(2^{-t}), & \text{for } i > r+1 \end{cases} . \tag{55}$$

From (54), we have that

$$H = N^{-1}(A - FN^{-1})N$$

and

$$N^{-1}AN = H + N^{-1}F. \qquad (56)$$

From (56), it is clear that H differs from an exact similarity transformation of the matrix A by the matrix $N^{-1}F$.

If the elements of A and H are not dissimilar in magnitude, we can assume that all their entries are bounded above by one, and so the h factors can be excluded from (55). Therefore, one source of danger happens when the elements of H are much greater than those of A, and, by (55), some of the bounds on the f_{ij} can be quite large. Another source of trouble occurs with the growth of the elements of N^{-1}. It is conceivable that the norm of N^{-1} could get unacceptably large because our transformation matrix N is not unitary. Apparently, no analysis of the growth of the elements of N^{-1} has been made.

The above analysis can be found in [6], where it is also stated that in practice, $||N^{-1}F||_2$ (Euclidian norm) has been usually found to be bounded above by $\frac{1}{2}n2^{-t}$. This should provide us with hope in terms of the usefulness of our reduction algorithm.

Remark: It should have been apparent from equations (50) and (51) that H was assumed to be in upper Hessenberg form. The error analysis for a lower Hessenberg matrix is the same with minor modifications.

A1.8 SUMMARY

To summarize this report, we shall briefly mention the different techniques that are contained herein and discuss the operation counts associated with each. Before doing so, we restate that our problem is to solve the equation

$$X' = AX,$$

where A is a general n x n sparce transition matrix with constant coefficients. Hence, we choose to solve this equation by finding the eigenvalues and eigenvectors of the matrix A and then expressing the general solution as a linear combination of the functions defined by equation (A). (See section A1.1.)

PROCEDURE FOR EIGENVALUES

I. Reduction to Hessenberg Form

1. If the Givens' method is used to reduce the matrix A to Hessenberg form, then

$$\frac{10}{3}n^3 + \text{(lower order terms)}$$

multiplications, and

$$\frac{5}{3}n^3 + \text{(lower order terms)}$$

additions to perform this procedure.

2. If Gaussian reduction is used, then

$$\frac{5}{6}n^3 + \text{(lower order terms)}$$

multiplications, and

$$\frac{5}{6}n^3 + \text{(lower order terms)}$$

additions are required to perform this procedure. Also, our algorithm for sparce reduction requires at most $\frac{5}{6}n^3$ + (lower order terms) checks on possible fill-in. These checks, however, do not involve any computations.

It is clear that Gaussian reduction requires one quarter the number of multiplications and one half the number of additions as the Givens' method. Even though we have a significant number of non-computational checks, it is safe to conclude that Gaussian reduction requires fewer operations than the Givens' scheme. When it is also seen that the number of fill-ins produced by the Givens' method is prohibitively large, it must be concluded that sparce Gaussian elimination is better in terms of time and use of sparcity.

II. Computation of Eigenvalues

The Hyman method is used to evaluate $p(\lambda)$ (see (9a)), $p'(\lambda)$ and $p''(\lambda)$ (see (47) and (47a)) in the implementation of the Newton-Raphson method and the Laguerre iteration scheme.

1. If the Newton-Raphson method is used (see (12)), then the computation of $p(\lambda)$ and $p'(\lambda)$ are required. To compute an eigenvalue by this process,

$$n^2 - n$$

multiplications, and

$$n^2 - n + 1$$

additions are required per iteration.

2. If the Laguerre technique is used (see (42)), then the computation of $p(\lambda)$, $p'(\lambda)$ and $p''(\lambda)$ are required. To compute an eigenvalue by this process,

$$n^2 + \text{(lower order terms)}$$

multiplications, and

$$n^2 + \text{(lower order terms)}$$

additions are required per iteration.

Apparently, there is very little difference in the number of operations per iteration between the two above methods. However, there are two distinct advantages that the Laguerre method has over Newton's method. Firstly, Laguerre's scheme usually converges cubically, whereas Newton's method converges quadratically. In [10], it is stated that less than an average of three iterations per eigenvalue (for Laguerre) were needed on a wide variety of matrices of orders from 8 to 100. Secondly, no a priori knowledge of the location of the eigenvalues is needed to start the Laguerre technique. (See section A1.7.7 on the selection of a starting value.) Thus, the Laguerre method seems to provide a distinct advantage over Newton's method.

In short, if we make the gross assumption that $0(n)$ iterations are required for convergence, it takes $0(n^3)$ multiplications and $0(n^3)$ additions to compute one eigenvalue using the Gaussian-Laguerre combination. The Givens'-Newton combination also requires $0(n^3)$ additions and multiplications, but where the constant multiple of n^3 is two or three times greater than that for Gaussian-Laguerre. Thus we conclude that the Gaussian-Laguerre method is the more viable of the two methods.

PROCEDURE FOR EIGENVECTORS

It is being assumed that the Weilandt iteration procedure,

$$(A - \lambda I)X_{n+1} = X_n,$$

together with Gaussian elimination is used to solve for an eigenvector that corresponds to λ.

Thus, to use this procedure

$$\frac{1}{2}n^2 + \frac{5}{2}n - 4$$

multiplications,

$$\frac{1}{2}n^2 + \frac{3}{2}n - 2$$

additions, and

$$n - 1$$

divisions are required for the first iteration, and

$$n - 1$$

multiplications, and

$$n - 1$$

additions are required for each subsequent iteration. In addition to this,

$$\frac{1}{2}n^2 - \frac{1}{2}n$$

multiplications, the same number of additions and n divisions are needed for the substitution process in each iteration. Finally, n^2 multiplications and $n^2 - n$ additions are needed to transform the computed eigenvector back to the original coordinate system.

Therefore, if $O(n)$ iterations are required for convergence, then $O(n^3)$ multiplications, $O(n^3)$ additions, and $O(n^2)$ divisions are needed to compute one eigenvector.

We conclude this report by stating that a Fortran IV computer program is being written to compute the eigenvalues and eigenvectors of the matrix A by means of sparce Gaussian reduction, the Laguerre iteration technique and the method of inverse iteration.

APPENDIX 2

COMPUTER PRINT-OUTS

A2.1 INTRODUCTION

Copies of a portion of the computer print-outs from some of of the reliability model runs are presented on the following pages. It should be noted that the complete print-outs are much more extensive, including, at the user's option, $P_\ell(t)$, $P^*_\ell(t)$ and $Q_\ell(t)$ for each state ℓ and for each time step $t = i\Delta t$. For brevity, only the summations $Q(t) = \sum_{\ell \varepsilon L} Q_\ell(t)$, $P^*(t) = \sum_{\ell \varepsilon L} P_\ell(t)$ and $1 - R(t) = Q(t) + P^*(t)$ are reproduced here.

OLTSUM =	0.	.4356765196E-08	.1411347080E-07	.2138572820E-07	.3093367852E-07	
	.3829726867E-07	.4764346985E-07	.5506700312E-07	.6421441994E-07	.7166696467E-07	
	.8061502460E-07	.8806640991E-07	.9681177154E-07	.1042328272E-06	.1127700324E-06	
	.1201327764E-06	.1284548023E-06	.1357325948E-06	.1438313110E-06	.1509989227E-06	
	.1580655242E-06	.1658592748E-06	.1735245532E-06	.1804023261E-06	.1877769864E-06	
	.1944752727E-06	.2015931548E-06	.2080990713E-06	.2149453410E-06	.2212386343E-06	
	.2278079411E-06	.2338729811E-06	.2401575661E-06	.2459803492E-06	.2519732276E-06	
	.2573412727E-06	.2632361945E-06	.2685386325E-06	.2739302238E-06	.2789576806E-06	
	.2840414652E-06	.2887860437E-06	.2935584849E-06	.2980137078E-06	.3024722080E-06	
	.3066329842E-06	.3107759544E-06	.3146384736E-06	.3184651187E-06	.3220269869E-06	.3255376[illegible]53E-06
P* SUM =	0.	.4759132931E-08	.5018474605E-07	.2406130927E-06	.7709503040E-06	
	.1941198226E-05	.4162045621E-05	.7993809726E-05	.1409299224E-04	.2327669381E-04	
	.3649511266E-04	.5483833559E-04	.7953561317E-04	.1119532960E-03	.1535915947E-03	
	.2060823126E-03	.2711736889E-03	.3507444771E-03	.4467773750E-03	.5613619097E-03	
	.6966848751E-03	.8559224089E-03	.1038731788E-02	.1250243017E-02	.1492050270E-02	
	.1766703255E-02	.2076798540E-02	.2424970906E-02	.2813884752E-02	.3246225610E-02	
	.3724391786E-02	.4251986172E-02	.4830808246E-02	.5463846298E-02	.6153769869E-02	
	.6903222479E-02	.7714814595E-02	.8591116899E-02	.9534653842E-02	.1054789751E-01	
	.1163326178E-01	.1279309681E-01	.1402966385E-01	.1534523036E-01	.1674186545E-01	
	.1822163562E-01	.1978650092E-01	.2143833128E-01	.2317890326E-01	.2500989707E-01	.2693289394E-01
G+P* SUM =	0.	.9115504128E-08	.6429821693E-07	.2619958209E-06	.8018839825E-06	
	.1979495497E-05	.4215689091E-05	.8048876730E-05	.1415720666E-04	.2334836077E-04	
	.3657572769E-04	.5492640200E-04	.7963242494E-04	.1120575289E-03	.1537043648E-03	
	.2062094454E-03	.2713021437E-03	.3509802097E-03	.4469212063E-03	.5615129086E-03	
	.6968437406E-03	.8551883282E-03	.1038905313E-02	.1250423419E-02	.1492238047E-02	
	.1766897733E-02	.2077000134E-02	.2425179005E-02	.2814099697E-02	.3246446848E-02	
	.3724919594E-02	.4252220045E-02	.4831048405E-02	.5464092278E-02	.6154021843E-02	
	.6903480020E-02	.7715077831E-02	.8591385437E-02	.9534927773E-02	.1054817647E-01	
	.1163354582E-01	.1279338559E-01	.1402997741E-01	.1534552838E-01	.1674216792E-01	
	.1822194226E-01	.1978681170E-01	.2143864592E-01	.2317922172E-01	.2501021910E-01	.2693321948E-01

Table A2-1

RM4, Recovery Rate Averaged, FTMP, permanent failures

T_{max} = 1000 hrs

N_p = 15, N_m = 8, N_B = 4

GLISUM	= 0.	.8284643018E-14	.3189783213E-13	.6616059009E-13	.1099410088E-12	
	.1598651937E-12	.2158079439E-12	.2751771381E-12	.3384546815E-12	.4035199210E-12	
	.4712208911E-12	.5397212705E-12	.6100898962E-12	.6806619205E-12	.7526393404E-12	
	.8244607205E-12	.8974083622E-12	.9699831962E-12	.1043515953E-11	.1116545175E-11	
	.1190430798E-11	.1263734049E-11	.1337832476E-11	.1411300985E-11	.1485527747E-11	
	.1559095919E-11	.1633400076E-11	.1707028351E-11	.1781379181E-11	.1855043701E-11	
	.1929422677E-11	.2003109055E-11	.2077505004E-11	.2151204563E-11	.2225610747E-11	
	.2299318254E-11	.2373730608E-11	.2447442908E-11	.2521858983E-11	.2595574171E-11	
	.2669992488E-11	.2743709418E-11	.2818129086E-11	.2891847065E-11	.2966267546E-11	
	.3039986157E-11	.3114407127E-11	.3188126118E-11	.3262547382E-11	.3336266602E-11	.3410688041E-11
P* SUM	= 0.	.4629482749E-28	.3703823046E-27	.1250013632E-26	.2962963694E-26	
	.5787001456E-26	.1000010904E-25	.1587970607E-25	.2370370952E-25	.3374988837E-25	
	.4629660369E-25	.6162056456E-25	.8000001954E-25	.1017127376E-24	.1270376483E-24	
	.1562503711E-24	.1896296757E-24	.2274533311E-24	.2700010246E-24	.3175469075E-24	
	.3703704598E-24	.4287494506E-24	.4929645146E-24	.5632879556E-24	.6400001537E-24	
	.7233768730E-24	.8137059000E-24	.9112512971E-24	.1016296539E-23	.1129119395E-23	
	.1250002962E-23	.1379214714E-23	.1517037397E-23	.1663748783E-23	.1819633483E-23	
	.1984755987E-23	.2160000519E-23	.2345044823E-23	.2540375244E-23	.2746252898E-23	
	.2962963659E-23	.3190785297E-23	.3430006026E-23	.3680883228E-23	.3943704625E-23	
	.4218747986E-23	.4506303610E-23	.4806624758E-23	.5120001189E-23	.5446710672E-23	.5787045779E-23
G+P* SUM	= 0.	.8284643018E-14	.3189783213E-13	.6616059009E-13	.1099410088E-12	
	.1598651937E-12	.2158079439E-12	.2751771381E-12	.3384546815E-12	.4035199210E-12	
	.4712208911E-12	.5397212705E-12	.6100898962E-12	.6806619205E-12	.7526393404E-12	
	.8244607205E-12	.8974083622E-12	.9699831962E-12	.1043515953E-11	.1116545175E-11	
	.1190430798E-11	.1263734049E-11	.1337832476E-11	.1411300985E-11	.1485527747E-11	
	.1559095919E-11	.1633400076E-11	.1707028351E-11	.1781379181E-11	.1855043701E-11	
	.1929422677E-11	.2003109055E-11	.2077505004E-11	.2151204563E-11	.2225610747E-11	
	.2299318254E-11	.2373730608E-11	.2447442908E-11	.2521858983E-11	.2595574171E-11	
	.2669992488E-11	.2743709418E-11	.2818129086E-11	.2891847065E-11	.2966267546E-11	
	.3039986157E-11	.3114407127E-11	.3188126118E-11	.3262547382E-11	.3336266602E-11	.3410688041E-11

Table A2-2

RM4, Recovery Rate Averaged, FTMP, permanent failures

T_{max} = 30 sec

N_p = 15, N_m = 9, N_B = 5

QLTSUM =	0.	.6686304258E-17	.2671039272E-16	.5993392193E-16	.1062918981E-15	
	.1656510437E-15	.2379508436E-15	.3230620100E-15	.4209275885E-15	.5314214922E-15	
	.6544896617E-15	.7900086312E-15	.9379267400E-15	.1098122732E-14	.1270546981E-14	
	.1455080124E-14	.1651674311E-14	.1860211842E-14	.2080646442E-14	.2312861908E-14	
	.2556813387E-14	.2812386050E-14	.3079536356E-14	.3358150751E-14	.3648186924E-14	
	.3949532523E-14	.4262146401E-14	.4585917347E-14	.4920805334E-14	.5266700244E-14	
	.5623563128E-14	.5991284928E-14	.6369827739E-14	.6759083532E-14	.7159015417E-14	
	.7569516370E-14	.7990550494E-14	.8422011747E-14	.8863865207E-14	.9316005793E-14	
	.9778399541E-14	.1025094231E-13	.1073360108E-13	.1122627264E-13	.1172892489E-13	
	.1224145553E-13	.1276383338E-13	.1329595703E-13	.1383779619E-13	.1438925037E-13	.1495029013E-13

P* SUM =	0.	.8800221654E-33	.7023321359E-32	.2372266245E-31	.5618657087E-31	
	.1097920375E-30	.1896296767E-30	.3012280753E-30	.4494925670E-30	.6401707042E-30	
	.8779151696E-30	.1168759852E-29	.1517037413E-29	.1929135448E-29	.2408999226E-29	
	.2963437419E-29	.3595940535E-29	.4313805693E-29	.5120001270E-29	.6022380199E-29	
	.7023321358E-29	.8131300865E-29	.9348040727E-29	.1068270762E-28	.1213629931E-28	
	.1371874039E-28	.1543023702E-28	.1728153910E-28	.1927199380E-28	.2141324369E-28	
	.2370370958E-28	.2615599408E-28	.2876752428E-28	.3155193019E-28	.3450557783E-28	
	.3764319196E-28	.4096001015E-28	.4447191932E-28	.4817296119E-28	.5208025219E-28	
	.5618657095E-28	.6051033050E-28	.6504297908E-28	.6980429418E-28	.7478432580E-28	
	.8000428316E-28	.8545275094E-28	.9115243736E-28	.9709039443E-28	.1032908967E-27	.1097393962E-27

Q+P* SUM =	0.	.6686304258E-17	.2671039272E-16	.5993392193E-16	.1062918981E-15	
	.1656510437E-15	.2379508436E-15	.3230620100E-15	.4209275885E-15	.5314214922E-15	
	.6544896617E-15	.7900086312E-15	.9379267400E-15	.1098122732E-14	.1270546981E-14	
	.1455080124E-14	.1651674311E-14	.1860211842E-14	.2080646442E-14	.2312861908E-14	
	.2556813387E-14	.2812386050E-14	.3079536356E-14	.3358150751E-14	.3648186924E-14	
	.3949532523E-14	.4262146401E-14	.4585917347E-14	.4920805334E-14	.5266700244E-14	
	.5623563128E-14	.5991284928E-14	.6369827739E-14	.6759083532E-14	.7159015417E-14	
	.7569516370E-14	.7990550494E-14	.8422011747E-14	.8863865207E-14	.9316005793E-14	
	.9778399541E-14	.1025094231E-13	.1073360108E-13	.1122627264E-13	.1172892489E-13	
	.1224145553E-13	.1276383338E-13	.1329595703E-13	.1383779619E-13	.1438925037E-13	.1495029013E-13

Table A2-3

RM4, Recovery Rate Averaged, FTMP, permanent failures

T_{max} = 800 msec

$N_p = 15$, $N_m = 9$, $N_B = 5$

GLTSUM	= 0.	-.9805489753E-17	.8497678733E-08	.1697350695E-07	.2542526311E-07	
	.3385032663E-07	.4224573096E-07	.5060821217E-07	.5893425349E-07	.6722012596E-07	
	.7546192560E-07	.8365560724E-07	.9179701551E-07	.9988191293E-07	.1079060055E-06	
	.1158649660E-06	.1237544552E-06	.1315701409E-06	.1393077151E-06	.1469629103E-06	
	.1545315130E-06	.1620093769E-06	.1693924340E-06	.1766767055E-06	.1838583104E-06	
	.1909334741E-06	.1978985357E-06	.2047499543E-06	.2114843148E-06	.2180983331E-06	
	.2245888600E-06	.2309528854E-06	.2371875410E-06	.2432901035E-06	.2492579962E-06	
	.2550887911E-06	.2607802101E-06	.2663301260E-06	.2717365627E-06	.2769976961E-06	
	.2821118534E-06	.2870775130E-06	.2918933038E-06	.2965580045E-06	.3010705421E-06	
	.3054299907E-06	.3096355703E-06	.3136866442E-06	.3175827179E-06	.3213234366E-06	.3249085830E-06
P* SUM	= 0.	.4759138931E-08	.5018474605E-07	.2406130927E-06	.7709503040E-06	
	.1941198228E-05	.4168045621E-05	.7993809726E-05	.1409299224E-04	.2327669381E-04	
	.3649511266E-04	.5483833559E-04	.7953561317E-04	.1119532960E-03	.1535915947E-03	
	.2060803126E-03	.2711736889E-03	.3507444771E-03	.4467773750E-03	.5613619097E-03	
	.6966848751E-03	.8550224089E-03	.1038731788E-02	.1250243017E-02	.1492050270E-02	
	.1766703255E-02	.2076798540E-02	.2424970906E-02	.2813884752E-02	.3246225610E-02	
	.3724691786E-02	.4251986172E-02	.4830808248E-02	.5463846298E-02	.6153769869E-02	
	.6903222479E-02	.7714814595E-02	.8591116899E-02	.9534653842E-02	.1054789751E-01	
	.1163326178E-01	.1279309681E-01	.1402968385E-01	.1534523036E-01	.1674186545E-01	
	.1822163562E-01	.1978650092E-01	.2143833128E-01	.2317890326E-01	.2500989707E-01	.2693289394E-01
Q+P* SUM	= 0.	.4759138922E-08	.5868242478E-07	.2575865996E-06	.7963755671E-06	
	.1975048555E-05	.4210291352E-05	.8044417939E-05	.1415192649E-04	.2334391393E-04	
	.3657057459E-04	.5492199120E-04	.7962741019E-04	.1120531779E-03	.1536995007E-03	
	.2061961776E-03	.2712974433E-03	.3508760472E-03	.4469166827E-03	.5615088726E-03	
	.6968394066E-03	.8551844183E-03	.1038901181E-02	.1250419694E-02	.1492234129E-02	
	.1766894189E-02	.2076996439E-02	.2425175656E-02	.2814096236E-02	.3246443708E-02	
	.3724916375E-02	.4252217125E-02	.4831045435E-02	.5464089588E-02	.6154019127E-02	
	.6903477568E-02	.7715075375E-02	.8591383229E-02	.9534925579E-02	.1054817451E-01	
	.1163354389E-01	.1279338389E-01	.1402997575E-01	.1534552692E-01	.1674216652E-01	
	.1822194105E-01	.1978681056E-01	.2143864496E-01	.2317922084E-01	.2501021840E-01	.2693321885E-01

Table A2-4

RM2, Difference Equation, 50 steps, FTMP, permanent failures

T_{max} = 1000 hrs

$N_p = 15$, $N_m = 8$, $N_B = 4$

QLTSUM	= 0.	-.8526512829E-22	.1656928244E-13	.4598564935E-13	.8537873975E-13	
	.1325195419E-12	.1856770529E-12	.2435070124E-12	.3049654962E-12	.3692418150E-12	
	.4357064043E-12	.5038703579E-12	.5733540017E-12	.6438624886E-12	.7151668466E-12	
	.7870892611E-12	.8594916446E-12	.9322667619E-12	.1005331336E-11	.1078620695E-11	
	.1152084618E-11	.1225684103E-11	.1299388862E-11	.1373175375E-11	.1447025375E-11	
	.1520924678E-11	.1594862268E-11	.1668829591E-11	.1742820003E-11	.1816828346E-11	
	.1890850612E-11	.1964883692E-11	.2038925169E-11	.2112973165E-11	.2187026225E-11	
	.2261083217E-11	.2335143262E-11	.2409205677E-11	.2483269933E-11	.2557335617E-11	
	.2631402412E-11	.2705470067E-11	.2779538391E-11	.2853607234E-11	.2927676479E-11	
	.3001746037E-11	.3075815837E-11	.3149885824E-11	.3223955957E-11	.3298026203E-11	.3372096536E-11
P* SUM	= 0.	.4629482749E-28	.3703823046E-27	.1250013632E-26	.2962963694E-26	
	.5787001456E-26	.1000010904E-25	.1587970607E-25	.2370370952E-25	.3374988837E-25	
	.4629660369E-25	.6162056456E-25	.8000001954E-25	.1017127376E-24	.1270376483E-24	
	.1562503711E-24	.1896296757E-24	.2274533311E-24	.2700010246E-24	.3175469075E-24	
	.3703704598E-24	.4287494506E-24	.4929645146E-24	.5632879556E-24	.6400001537E-24	
	.7233788780E-24	.8137059000E-24	.9112512971E-24	.1016296539E-23	.1129119395E-23	
	.1250002962E-23	.1379214714E-23	.1517037397E-23	.1663748783E-23	.1819633483E-23	
	.1984955987E-23	.2160000510E-23	.2345044823E-23	.2540375244E-23	.2746252898E-23	
	.2962963659E-23	.3190785297E-23	.3430006026E-23	.3680883228E-23	.3943704625E-23	
	.4218747986E-23	.4506303610E-23	.4806624758E-23	.5120001189E-23	.5446710672E-23	.5787045779E-23
Q+P* SUM	= 0.	-.8526508200E-22	.1656928244E-13	.4598564935E-13	.8537873975E-13	
	.1325195419E-12	.1856770529E-12	.2435070124E-12	.3049654962E-12	.3692418150E-12	
	.4357064043E-12	.5038703579E-12	.5733540017E-12	.6438624886E-12	.7151668466E-12	
	.7870892611E-12	.8594916446E-12	.9322667619E-12	.1005331336E-11	.1078620695E-11	
	.1152084618E-11	.1225684103E-11	.1299388862E-11	.1373175375E-11	.1447025375E-11	
	.1520924678E-11	.1594862268E-11	.1668829591E-11	.1742820003E-11	.1816828346E-11	
	.1890850612E-11	.1964883692E-11	.2038925169E-11	.2112973165E-11	.2187026225E-11	
	.2261083217E-11	.2335143262E-11	.2409205677E-11	.2483269933E-11	.2557335617E-11	
	.2631402412E-11	.2705470067E-11	.2779538391E-11	.2853607234E-11	.2927676479E-11	
	.3001746037E-11	.3075815837E-11	.3149885824E-11	.3223955957E-11	.3298026203E-11	.3372096536E-11

Table A2-5
RM2, Difference Equation, 50 steps, FTMP, permanent failures

T_{max} = 30 sec

N_p = 15, N_m = 9, N_B = 5

QLTSUM	= 0.	0.	.1337260844E-16	.4001335283E-16	.7982061983E-16	
	.1326953235E-15	.1985406071E-15	.2772616055E-15	.3687652272E-15	.4729599766E-15	
	.5897558040E-15	.7190639648E-15	.8607969112E-15	.1014868200E-14	.1181192401E-14	
	.1359685030E-14	.1550262490E-14	.1752842012E-14	.1967341616E-14	.2193680070E-14	
	.2431776849E-14	.2681552117E-14	.2942926691E-14	.3215822030E-14	.3500160203E-14	
	.3795863880E-14	.4102856314E-14	.4421061330E-14	.4750403308E-14	.5090807178E-14	
	.5442198409E-14	.5804502997E-14	.6177647464E-14	.6561558842E-14	.6956164674E-14	
	.7361393003E-14	.7777172370E-14	.8203431805E-14	.8640100829E-14	.9087109438E-14	
	.9544388114E-14	.1001186780E-13	.1048947993E-13	.1097715637E-13	.1147482947E-13	
	.1198243204E-13	.1249989733E-13	.1302715905E-13	.1356415136E-13	.1411080886E-13	.1466706658E-13
P* SUM	= 0.	.8800221654E-33	.7023321359E-32	.2372266245E-31	.5618657087E-31	
	.1097920375E-30	.1896296767E-30	.3012280753E-30	.4494925670E-30	.6401707042E-30	
	.8779151698E-30	.1168759852E-29	.1517037413E-29	.1929135448E-29	.2408999226E-29	
	.2963437419E-29	.3595940535E-29	.4313805693E-29	.5120001270E-29	.6022380199E-29	
	.7023321358E-29	.8131300865E-29	.9348040727E-29	.1068270762E-28	.1213629931E-28	
	.1371874039E-28	.1543023702E-28	.1728153910E-28	.1927199380E-28	.2141324369E-28	
	.2370370958E-28	.2615599408E-28	.2876752428E-28	.3155193019E-28	.3450557783E-28	
	.3764319196E-28	.4096001015E-28	.4447191932E-28	.4817296119E-28	.5208025219E-28	
	.5618657085E-28	.6051033050E-28	.6504297908E-28	.6980429418E-28	.7478432580E-28	
	.8000428316E-28	.8545275094E-28	.9115243736E-28	.9709039443E-28	.1032908967E-27	.1097393962E-27
Q+P* SUM	= 0.	.8800221654E-33	.1337260844E-16	.4001335283E-16	.7982061983E-16	
	.1326953235E-15	.1985406071E-15	.2772616055E-15	.3687652272E-15	.4729599766E-15	
	.5897558040E-15	.7190639648E-15	.8607969112E-15	.1014868200E-14	.1181192401E-14	
	.1359685030E-14	.1550262490E-14	.1752842012E-14	.1967341616E-14	.2193680070E-14	
	.2431776849E-14	.2681552117E-14	.2942926691E-14	.3215822030E-14	.3500160203E-14	
	.3795863880E-14	.4102856314E-14	.4421061330E-14	.4750403308E-14	.5090807178E-14	
	.5442198409E-14	.5804502997E-14	.6177647464E-14	.6561558842E-14	.6956164674E-14	
	.7361393003E-14	.7777172370E-14	.8203431805E-14	.8640100829E-14	.9087109438E-14	
	.9544388114E-14	.1001186780E-13	.1048947993E-13	.1097715637E-13	.1147482947E-13	
	.1198243204E-13	.1249989733E-13	.1302715905E-13	.1356415136E-13	.1411080886E-13	.1466706658E-13

Table A2-6

RM2, Difference Equation, 50 steps, FTMP, permanent failures

$T_{max} = 800$

$N_p = 15$, $N_m = 9$, $N_B = 5$

	.1698357059E-07	.2121504507E-07	.2544014095E-07	.2965851159E-07	.3386978953E-07	
	.3807358794E-07	.4226950213E-07	.4645711087E-07	.5063597778E-07	.5480565251E-07	
	.5896567200E-07	.6311556164E-07	.6725483633E-07	.7138300157E-07	.7549955443E-07	
	.7960398457E-07	.8369577509E-07	.8777440346E-07	.9183934232E-07	.9589006029E-07	
	.9992602275E-07	.1039466925E-06	.1079515306E-06	.1119399969E-06	.1159115505E-06	
	.1198656508E-06	.1238017578E-06	.1277193325E-06	.1316178377E-06	.1354967384E-06	
	.1393555020E-06	.1431935992E-06	.1470105042E-06	.1508056950E-06	.1545786539E-06	
	.1583288679E-06	.1620558289E-06	.1657590342E-06	.1694379867E-06	.1730921950E-06	
	.1767211742E-06	.1803244457E-06	.1839015374E-06	.1874519843E-06	.1909753287E-06	
	.1944711198E-06	.1979389147E-06	.2013782781E-06	.2047887824E-06	.2081700083E-06	
	.2115215445E-06	.2148429881E-06	.2181339446E-06	.2213940281E-06	.2246228614E-06	
	.2278200759E-06	.2309853121E-06	.2341182194E-06	.2372184560E-06	.2402856895E-06	
	.2433195964E-06	.2463198627E-06	.2492861834E-06	.2522182629E-06	.2551158149E-06	
	.2579785626E-06	.2608062385E-06	.2635985845E-06	.2663553518E-06	.2690763014E-06	
	.2717612033E-06	.2744098373E-06	.2770219924E-06	.2795974670E-06	.2821360692E-06	
	.2846376161E-06	.2871019344E-06	.2895288600E-06	.2919182383E-06	.2942699237E-06	
	.2965837800E-06	.2988596602E-06	.3010975063E-06	.3032971495E-06	.3054585100E-06	
	.3075814970E-06	.3096660287E-06	.3117120320E-06	.3137194428E-06	.3156882055E-06	
	.3176182734E-06	.3195096083E-06	.3213621805E-06	.3231759688E-06	.3249509604E-06	.3266871506E-06
P* SUM =	0.	.7416532601E-09	.4759138931E-08	.1779874211E-07	.5018474605E-07	
	.1173999557E-06	.2406130927E-06	.4471556002E-06	.7709503040E-06	.1252894285E-05	
	.1941198228E-05	.2891684441E-05	.4168045621E-05	.5842066421E-05	.7993809726E-05	
	.1071176954E-04	.1409299224E-04	.1824316797E-04	.2327669381E-04	.2931671025E-04	
	.3649511266E-04	.4495253905E-04	.5483833559E-04	.6631050132E-04	.7953561317E-04	
	.9468873279E-04	.1119532960E-03	.1315209866E-03	.1535915947E-03	.1783728618E-03	
	.2060803126E-03	.2369370749E-03	.2711736889E-03	.3090279056E-03	.3507444771E-03	
	.3965749372E-03	.4467773750E-03	.5016162015E-03	.5613619097E-03	.6262908290E-03	
	.6966848751E-03	.7728312951E-03	.8550224089E-03	.9435553474E-03	.1038731788E-02	
	.1140857688E-02	.1250243017E-02	.1367201484E-02	.1492050270E-02	.1625109758E-02	
	.1766703255E-02	.1917156731E-02	.2076798540E-02	.2245959157E-02	.2424970906E-02	
	.2614167695E-02	.2813884752E-02	.3024458363E-02	.3246225610E-02	.3479524115E-02	
	.3724691786E-02	.3982066563E-02	.4251986172E-02	.4534787879E-02	.4830808248E-02	
	.5140382903E-02	.5463846298E-02	.5801531480E-02	.6153769869E-02	.6520891035E-02	
	.6903222479E-02	.7301089422E-02	.7714814595E-02	.8144718038E-02	.8591116899E-02	
	.9054325238E-02	.9534653842E-02	.1003241004E-01	.1054789751E-01	.1108141613E-01	
	.1163326178E-01	.1220372619E-01	.1279309681E-01	.1340165658E-01	.1402968385E-01	
	.1467745222E-01	.1534523036E-01	.1603328194E-01	.1674186545E-01	.1747123407E-01	
	.1822163562E-01	.1899331236E-01	.1978650092E-01	.2060143221E-01	.2143833128E-01	
	.2229741726E-01	.2317890326E-01	.2408299626E-01	.2500989707E-01	.2595980023E-01	.2693289394E-01
Q+P* SUM =	0.	.7416532552E-09	.9013134766E-08	.2630150080E-07	.6293078779E-07	
	.1343835263E-06	.2618281377E-06	.4725957411E-06	.8006088156E-06	.1286764074E-05	
	.1979271816E-05	.2933953943E-05	.4214502732E-05	.5892702398E-05	.8048615379E-05	
	.1077073521E-04	.1415610780E-04	.1831042281E-04	.2334807681E-04	.2939220980E-04	
	.3657471665E-04	.4503623483E-04	.5492611000E-04	.6640234066E-04	.7963150323E-04	
	.9478865881E-04	.1120572427E-03	.1316289381E-03	.1537035347E-03	.1784887734E-03	
	.2062001783E-03	.2370608767E-03	.2713014082E-03	.3091595234E-03	.3508799738E-03	
	.3967142927E-03	.4469205686E-03	.5017632120E-03	.5615127154E-03	.6264454077E-03	
	.6968432040E-03	.7729933509E-03	.8551881679E-03	.9437247854E-03	.1038904880E-02	
	.1141034410E-02	.1250423341E-02	.1367385385E-02	.1492237722E-02	.1625300733E-02	
	.1766897726E-02	.1917354670E-02	.2076999919E-02	.2246163946E-02	.2425179076E-02	
	.2614379216E-02	.2814099595E-02	.3024676497E-02	.3246447004E-02	.3479748738E-02	
	.3724919606E-02	.3982297548E-02	.4252220290E-02	.4535025097E-02	.4831048533E-02	
	.5140626223E-02	.5464092618E-02	.5801780766E-02	.6154022088E-02	.6521146151E-02	
	.6903480458E-02	.7301350228E-02	.7715078193E-02	.8144984393E-02	.8591385975E-02	
	.9054596999E-02	.9534928252E-02	.1003268706E-01	.1054817711E-01	.1108169826E-01	
	.1163354641E-01	.1220401330E-01	.1279338634E-01	.1340194850E-01	.1402997812E-01	
	.1467774880E-01	.1534552922E-01	.1603358304E-01	.1674216874E-01	.1747153953E-01	
	.1822194320E-01	.1899362203E-01	.1978681263E-01	.2060174593E-01	.2143864697E-01	
	.2229773488E-01	.2317922276E-01	.2408331762E-01	.2501022025E-01	.2596012519E-01	.2693322063E-01

Table A2-7

RM2, Difference Equation, 100 steps T_{max} = 1000 hrs N_p = 15, N_m = 8, N_B = 4

FTMP, permanent failures

QLTSUM	= 0.	-.4263256415E-22	.4409338212E-14	.1269397948E-13	.2439316532E-13	
	.3910134877E-13	.5646117127E-13	.7615771647E-13	.9791346828E-13	.1214838694E-12	
	.1466534067E-12	.1732321622E-12	.2010527730E-12	.2299677528E-12	.2598471310E-12	
	.2905763729E-12	.3220545461E-12	.3541927055E-12	.3869124690E-12	.4201447637E-12	
	.4538287198E-12	.4879106966E-12	.5223434241E-12	.5570852460E-12	.5920994533E-12	
	.6273536968E-12	.6628194689E-12	.6984716480E-12	.7342880956E-12	.7702493029E-12	
	.8063380776E-12	.8425392693E-12	.8788395272E-12	.9152270858E-12	.9516915770E-12	
	.9882238640E-12	.1024815895E-11	.1061460575E-11	.1098151650E-11	.1134883612E-11	
	.1171651603E-11	.1208451346E-11	.1245279068E-11	.1282131448E-11	.1319005556E-11	
	.1355898812E-11	.1392808942E-11	.1429733942E-11	.1466672046E-11	.1503621698E-11	
	.1540581525E-11	.1577550320E-11	.1614527018E-11	.1651510679E-11	.1688500478E-11	
	.1725495684E-11	.1762495656E-11	.1799499827E-11	.1836507699E-11	.1873518833E-11	
	.1910532840E-11	.1947549380E-11	.1984568151E-11	.2021588890E-11	.2058611361E-11	
	.2095635359E-11	.2132660703E-11	.2169687233E-11	.2206714808E-11	.2243743304E-11	
	.2280772612E-11	.2317802634E-11	.2354833286E-11	.2391864494E-11	.2428896191E-11	
	.2465928318E-11	.2502960826E-11	.2539993669E-11	.2577026806E-11	.2614060203E-11	
	.2651093829E-11	.2688127657E-11	.2725161662E-11	.2762195824E-11	.2799230124E-11	
	.2836264545E-11	.2873299074E-11	.2910333696E-11	.2947368402E-11	.2984403181E-11	
	.3021438024E-11	.3058472924E-11	.3095507874E-11	.3132542868E-11	.3169577900E-11	
	.3206612967E-11	.3243648063E-11	.3280683186E-11	.3317718332E-11	.3354753498E-11	.3391788683E-11

P* SUM	= 0.	.5786853437E-29	.4629482749E-28	.1562583655E-27	.3703823046E-27	
	.7233936865E-27	.1250013632E-26	.1984963261E-26	.2962963694E-26	.4218736053E-26	
	.5787001456E-26	.7702660142E-26	.1000010904E-25	.1271421731E-25	.1587970607E-25	
	.1953129643E-25	.2370370952E-25	.2843166646E-25	.3374988837E-25	.3969363076E-25	
	.4629660369E-25	.5359400792E-25	.6162056456E-25	.7041099472E-25	.8000001954E-25	
	.9042236012E-25	.1017127376E-24	.1139069522E-24	.1270376483E-24	.1411405475E-24	
	.1562503711E-24	.1724018401E-24	.1896296757E-24	.2079685990E-24	.2274533311E-24	
	.2481204065E-24	.2700010246E-24	.2931316179E-24	.3175469075E-24	.3432816144E-24	
	.3703704598E-24	.3988481649E-24	.4287494506E-24	.4601117753E-24	.4929645146E-24	
	.5273450010E-24	.5632879556E-24	.6008280995E-24	.6400001537E-24	.6808388395E-24	
	.7233788780E-24	.7676588404E-24	.8137059000E-24	.8615584785E-24	.9112512971E-24	
	.9628190769E-24	.1016296539E-23	.1071718405E-23	.1129119395E-23	.1188539384E-23	
	.1250002962E-23	.1313549832E-23	.1379214714E-23	.1447032328E-23	.1517037397E-23	
	.1589264642E-23	.1663748783E-23	.1740531186E-23	.1819633483E-23	.1901096843E-23	
	.1984955987E-23	.2071245635E-23	.2160000510E-23	.2251255332E-23	.2345044823E-23	
	.2441412030E-23	.2540375244E-23	.2641977293E-23	.2746252898E-23	.2853236779E-23	
	.2962963659E-23	.3075468258E-23	.3190785297E-23	.3308959695E-23	.3430006026E-23	
	.3553968963E-23	.3680883228E-23	.3810783541E-23	.3943704625E-23	.4079681199E-23	
	.4218747986E-23	.4360951965E-23	.4506303610E-23	.4654849634E-23	.4806624758E-23	
	.4961663703E-23	.5120001189E-23	.5281671938E-23	.5446710672E-23	.5615166619E-23	.5787045779E-23

Q+P* SUM	= 0.	-.4263255836E-22	.4409338212E-14	.1269397948E-13	.2439316532E-13	
	.3910134877E-13	.5646117127E-13	.7615771647E-13	.9791346828E-13	.1214838694E-12	
	.1466534067E-12	.1732321622E-12	.2010527730E-12	.2299677528E-12	.2598471310E-12	
	.2905763729E-12	.3220545461E-12	.3541927055E-12	.3869124690E-12	.4201447637E-12	
	.4538287198E-12	.4879106966E-12	.5223434241E-12	.5570852460E-12	.5920994533E-12	
	.6273536968E-12	.6628194689E-12	.6984716480E-12	.7342880956E-12	.7702493029E-12	
	.8063380776E-12	.8425392693E-12	.8788395272E-12	.9152270858E-12	.9516915770E-12	
	.9882238640E-12	.1024815895E-11	.1061460575E-11	.1098151650E-11	.1134883612E-11	
	.1171651603E-11	.1208451346E-11	.1245279068E-11	.1282131448E-11	.1319005556E-11	
	.1355898812E-11	.1392808942E-11	.1429733942E-11	.1466672046E-11	.1503621698E-11	
	.1540581525E-11	.1577550320E-11	.1614527018E-11	.1651510679E-11	.1688500478E-11	
	.1725495684E-11	.1762495656E-11	.1799499827E-11	.1836507699E-11	.1873518833E-11	
	.1910532840E-11	.1947549380E-11	.1984568151E-11	.2021588890E-11	.2058611361E-11	
	.2095635359E-11	.2132660703E-11	.2169687233E-11	.2206714808E-11	.2243743304E-11	
	.2280772612E-11	.2317802634E-11	.2354833286E-11	.2391864494E-11	.2428896191E-11	
	.2465928318E-11	.2502960826E-11	.2539993669E-11	.2577026806E-11	.2614060203E-11	
	.2651093829E-11	.2688127657E-11	.2725161662E-11	.2762195824E-11	.2799230124E-11	
	.2836264545E-11	.2873299074E-11	.2910333696E-11	.2947368402E-11	.2984403181E-11	
	.3021438024E-11	.3058472924E-11	.3095507874E-11	.3132542868E-11	.3169577900E-11	
	.3206612967E-11	.3243648063E-11	.3280683186E-11	.3317718332E-11	.3354753498E-11	.3391788683E-11

Table A2-8

RM2, Difference Equation, 100 steps T_{max} = 30 sec N_p = 15, N_m = 9, N_B = 5

FTMP, permanent failures

QLTSUM =	0.	0.	.3349778148E-17	.1003608237E-16	.2004585583E-16	
	.3336622803E-16	.4998449879E-16	.6988813229E-16	.9306474503E-16	.1195020969E-15	
	.1491880826E-15	.1821107244E-15	.2182581658E-15	.2576186650E-15	.3001805867E-15	
	.3459323976E-15	.3948626621E-15	.4469600368E-15	.5022132667E-15	.5606111804E-15	
	.6221426848E-15	.6867967652E-15	.7545624786E-15	.8254289518E-15	.8993853809E-15	
	.9764210251E-15	.1056525206E-14	.1139687307E-14	.1225896766E-14	.1315143081E-14	
	.1407415801E-14	.1502704531E-14	.1600998924E-14	.1702288686E-14	.1806563568E-14	
	.1913813370E-14	.2024027939E-14	.2137197166E-14	.2253310985E-14	.2372359374E-14	
	.2494332354E-14	.2619219988E-14	.2747012377E-14	.2877699665E-14	.3011272033E-14	
	.3147719702E-14	.3287032931E-14	.3429202017E-14	.3574217294E-14	.3722069133E-14	
	.3872747939E-14	.4026244156E-14	.4182548261E-14	.4341650769E-14	.4503542227E-14	
	.4668213216E-14	.4835654353E-14	.5005856289E-14	.5178809706E-14	.5354505321E-14	
	.5532933885E-14	.5714086179E-14	.5897953019E-14	.6084525253E-14	.6273793758E-14	
	.6465749447E-14	.6660383261E-14	.6857686177E-14	.7057649198E-14	.7260263363E-14	
	.7465519737E-14	.7673409420E-14	.7883923541E-14	.8097053259E-14	.8312789764E-14	
	.8531124275E-14	.8752048044E-14	.8975552348E-14	.9201628500E-14	.9430267838E-14	
	.9661461730E-14	.9895201575E-14	.1013147880E-13	.1037028486E-13	.1061161125E-13	
	.1085544947E-13	.1110179107E-13	.1135062762E-13	.1160195072E-13	.1185575200E-13	
	.1211202313E-13	.1237075577E-13	.1263194165E-13	.1289557252E-13	.1316164013E-13	
	.1343013628E-13	.1370105281E-13	.1397438155E-13	.1425011440E-13	.1452824326E-13	.1480876007E-13

P* SUM =	0.	.1100027707E-33	.8800221654E-33	.2960595853E-32	.7023321359E-32	
	.1372400468E-31	.2372266245E-31	.3762771934E-31	.5618657087E-31	.8002133803E-31	
	.1097920375E-30	.1460312959E-30	.1896296767E-30	.2411419310E-30	.3012280753E-30	
	.3703112535E-30	.4494925670E-30	.5392257117E-30	.6401707042E-30	.7526075205E-30	
	.8779151698E-30	.1016412608E-29	.1168759852E-29	.1335060025E-29	.1517037413E-29	
	.1714842549E-29	.1929135448E-29	.2159808695E-29	.2408999226E-29	.2676655462E-29	
	.2963437419E-29	.3268993459E-29	.3595940535E-29	.3943991275E-29	.4313805693E-29	
	.4704754244E-29	.5120001270E-29	.5558989916E-29	.6022380199E-29	.6509230980E-29	
	.7023321358E-29	.7563791314E-29	.8131300865E-29	.8724563594E-29	.9348040727E-29	
	.1000053540E-28	.1068270762E-28	.1139289201E-28	.1213629931E-28	.1291136209E-28	
	.1371874039E-28	.1455635617E-28	.1543023702E-28	.1633841133E-28	.1728153910E-28	
	.1825709599E-28	.1927199380E-28	.2032382303E-28	.2141324369E-28	.2253725139E-28	
	.2370370958E-28	.2490973714E-28	.2615599408E-28	.2743896232E-28	.2876752428E-28	
	.3013829356E-28	.3155193019E-28	.3300436869E-28	.3450557783E-28	.3605163225E-28	
	.3764319196E-28	.3927561044E-28	.4096001015E-28	.4269189311E-28	.4447191932E-28	
	.4629482749E-28	.4817296119E-28	.5010121608E-28	.5208025219E-28	.5410415977E-28	
	.5618657085E-28	.5832174109E-28	.6051033050E-28	.6274574721E-28	.6504297908E-28	
	.6739560807E-28	.6980429418E-28	.7226172974E-28	.7478432580E-28	.7736495694E-28	
	.8000428316E-28	.8269424729E-28	.8545275094E-28	.8827192763E-28	.9115243736E-28	
	.9408543977E-28	.9709039443E-28	.1001586601E-27	.1032908967E-27	.1064774471E-27	.1097393962E-27

Q+P* SUM =	0.	.1100027707E-33	.3349778148E-17	.1003608237E-16	.2004585583E-16	
	.3336622803E-16	.4998449879E-16	.6988813229E-16	.9306474503E-16	.1195020969E-15	
	.1491880826E-15	.1821107244E-15	.2182581658E-15	.2576186650E-15	.3001805867E-15	
	.3459323976E-15	.3948626621E-15	.4469600368E-15	.5022132667E-15	.5606111804E-15	
	.6221426848E-15	.6867967652E-15	.7545624786E-15	.8254289518E-15	.8993853809E-15	
	.9764210251E-15	.1056525206E-14	.1139687307E-14	.1225896766E-14	.1315143081E-14	
	.1407415801E-14	.1502704531E-14	.1600998924E-14	.1702288686E-14	.1806563568E-14	
	.1913813370E-14	.2024027939E-14	.2137197166E-14	.2253310985E-14	.2372359374E-14	
	.2494332354E-14	.2619219988E-14	.2747012377E-14	.2877699665E-14	.3011272033E-14	
	.3147719702E-14	.3287032931E-14	.3429202017E-14	.3574217294E-14	.3722069133E-14	
	.3872747939E-14	.4026244156E-14	.4182548261E-14	.4341650769E-14	.4503542227E-14	
	.4668213216E-14	.4835654353E-14	.5005856289E-14	.5178809706E-14	.5354505321E-14	
	.5532933885E-14	.5714086179E-14	.5897953019E-14	.6084525253E-14	.6273793758E-14	
	.6465749447E-14	.6660383261E-14	.6857686177E-14	.7057649198E-14	.7260263363E-14	
	.7465519737E-14	.7673409420E-14	.7883923541E-14	.8097053259E-14	.8312789764E-14	
	.8531124275E-14	.8752048044E-14	.8975552348E-14	.9201628500E-14	.9430267838E-14	
	.9661461730E-14	.9895201575E-14	.1013147880E-13	.1037028486E-13	.1061161125E-13	
	.1085544947E-13	.1110179107E-13	.1135062762E-13	.1160195072E-13	.1185575200E-13	
	.1211202313E-13	.1237075577E-13	.1263194165E-13	.1289557252E-13	.1316164013E-13	
	.1343013628E-13	.1370105281E-13	.1397438155E-13	.1425011440E-13	.1452824326E-13	.1480876007E-13

Table A2-9

RM2, Difference Equation, 100 steps $T_{max} = 800$ msec $N_p = 15$, $N_m = 9$, $N_B = 5$

FTMP, permanent failures

QLTSUM =	0.	.4551770786E-08	.1472789133E-07	.2233366275E-07	.3229673499E-07	
	.4002084649E-07	.4979310213E-07	.5761516367E-07	.6721602603E-07	.7511826810E-07	
	.8456470420E-07	.9253164425E-07	.1018384716E-06	.1098566208E-06	.1190367742E-06	
	.1270943789E-06	.1361591421E-06	.1442459568F-06	.1532051633F-06	.1613122494E-06	
	.1701744563E-06	.1782940049E-06	.1870666436F-06	.1951918190E-06	.2038813245E-06	
	.2120061259E-06	.2206180490E-06	.2287371881F-06	.2372762918E-06	.2453850845E-06	
	.2538554273E-06	.2619496985E-06	.2703547058E-06	.2784307047E-06	.2867732305E-06	
	.2948275566E-06	.3031099369E-06	.3111394745F-06	.3193635732E-06	.3273654338E-06	
	.3355326825E-06	.3435041548E-06	.3516155879E-06	.3595540934E-06	.3676103780E-06	
	.3755134336E-06	.3835148968E-06	.3913800813F-06	.3993267336E-06	.4071516600E-06	.4150432164E-06
P* SUM =	0.	.6853144684E-17	.1525528285F-14	.3791191622E-13	.3692577483E-12	
	.2147930693E-11	.9015168948E-11	.3020543503E-10	.8581700088E-10	.2149590339E-09	
	.4875114262F-09	.1020237507E-08	.1997974982E-08	.3700599994F-08	.6536415855E-08	
	.1108256414E-07	.1813299380E-07	.2875445533F-07	.4435091432E-07	.6673670203E-07	
	.9821864330E-07	.1416873231E-06	.2007175755E-06	.2796782007E-06	.3838508437E-06	
	.5195578927E-06	.6942991887E-06	.9168972719E-06	.1197650829E-05	.1548495950E-05	
	.1983174741E-05	.2517410818E-05	.3169091117E-05	.3958453472E-05	.4908279330E-05	
	.6044090978E-05	.7394352610E-05	.8990674545E-05	.1086801991E-04	.1306491308E-04	
	.1562364909E-04	.1859050341E-04	.2201594127E-04	.2595482583E-04	.3046662450E-04	
	.3561561271E-04	.4147107437E-04	.4810749850E-04	.5560477115E-04	.6404836219E-04	.7352950616E-04
Q+P* SUM =	0.	.4551770793F-08	.1472789286F-07	.2233370066E-07	.3229710425E-07	
	.4002299442E-07	.4980211730E-07	.5764536911E-07	.6730184303E-07	.7533322713E-07	
	.8505221562E-07	.9355188176E-07	.1038364466E-06	.1135572208E-06	.1255731900E-06	
	.1381769431E-06	.1542921359E-06	.1730004122E-06	.1975560776E-06	.2280489514E-06	
	.2683930996E-06	.3199813280E-06	.3877842190E-06	.4748700197E-06	.5877321682E-06	
	.7315640187E-06	.9149172377E-06	.1145634460E-05	.1434927121E-05	.1793881034E-05	
	.2237030169E-05	.2779360517E-05	.3439445823F-05	.4236884177E-05	.5195052560E-05	
	.6338918535E-05	.7697462546E-05	.9301814019E-05	.1118738349E-04	.1339227851E-04	
	.1595918177E-04	.1893400756E-04	.2236755686E-04	.2631437992F-04	.3083423488F-04	
	.3599112614E-04	.4185458927E-04	.4849887858F-04	.5600409789E-04	.6445551385E-04	.7394454938E-04

Table A2-10

Form 1, Recovery Distribution Averaged, FTMP, permanent failures

T_{max} = 1000 hrs

N_p = 1[illegible], N_m = [illegible], N_B = 5

QLTSUM =	0.	.4445293693E-10	.1481320702E-09	.2222244882E-09	.3258789051E-09	
	.3999876354E-09	.5036173641E-09	.5777423787E-09	.6813474476E-09	.7554887186E-09	
	.8590691560E-09	.9332266556E-09	.1036782490E-08	.1110956190E-08	.1214487449E-08	
	.1288677322E-08	.1392184034E-08	.1466390053E-08	.1569872246E-08	.1644094382E-08	
	.1747552084E-08	.1821790311E-08	.1925223550E-08	.1999477839E-08	.2102886643E-08	
	.2177156967E-08	.2280541364E-08	.2354827695E-08	.2458187714E-08	.2532490024E-08	
	.2635825692E-08	.2710143954E-08	.2813455299E-08	.2887789486E-08	.2991076535E-08	
	.3065426620E-08	.3168689401E-08	.3243055356E-08	.3346293898E-08	.3420675695E-08	
	.3523890024E-08	.3598287637E-08	.3701477782E-08	.3775891183E-08	.3879057171E-08	
	.3953486334E-08	.4056628192E-08	.4131073088E-08	.4234190845E-08	.4308651448E-08	.4411745130E-08

P* SUM =	0.	.7999995363E-26	.1279998455E-24	.6479988434E-24	.2047995110E-23	
	.4999985002E-23	.1036796275E-22	.1920791941E-22	.3276784258E-22	.5248771653E-22	
	.7999951973E-22	.1171272273E-21	.1658868061E-21	.2284862177E-21	.3073254194E-21	
	.4049963560E-21	.5242829661E-21	.6681611850E-21	.8397989295E-21	.1042556118E-20	
	.1279984644E-20	.1555828399E-20	.1874023270E-20	.2238697114E-20	.2654169783E-20	
	.3124953133E-20	.3655750976E-20	.4251459122E-20	.4917165390E-20	.5658149568E-20	
	.6479833365E-20	.7388030587E-20	.8388446934E-20	.9487180157E-20	.1069046991E-19	
	.1200474793E-19	.1343663781E-19	.1499295518E-19	.1668070767E-19	.1850709493E-19	
	.2047950854E-19	.2260553190E-19	.2489294069E-19	.2734970242E-19	.2998397639E-19	
	.3280411434E-19	.3581865941E-19	.3903634719E-19	.4246610495E-19	.4611705218E-19	.4999850007E-19

Q+P* SUM =	0.	.4445293693E-10	.1481320702E-09	.2222244882E-09	.3258789051E-09	
	.3999876354E-09	.5036173641E-09	.5777423787E-09	.6813474476E-09	.7554887186E-09	
	.8590691560E-09	.9332266556E-09	.1036782490E-08	.1110956190E-08	.1214487449E-08	
	.1288677322E-08	.1392184034E-08	.1466390053E-08	.1569872246E-08	.1644094382E-08	
	.1747552084E-08	.1821790311E-08	.1925223550E-08	.1999477839E-08	.2102886643E-08	
	.2177156967E-08	.2280541364E-08	.2354827695E-08	.2458187714E-08	.2532490024E-08	
	.2635825692E-08	.2710143954E-08	.2813455299E-08	.2887789486E-08	.2991076535E-08	
	.3065426620E-08	.3168689401E-08	.3243055356E-08	.3346293898E-08	.3420675695E-08	
	.3523890024E-08	.3598287637E-08	.3701477782E-08	.3775891183E-08	.3879057171E-08	
	.3953486334E-08	.4056628192E-08	.4131073088E-08	.4234190845E-08	.4308651448E-08	.4411745130E-08

Table A2-11

Form 1, Recovery Distribution Averaged, FTMP, permanent failures

T_{max} = 10 hrs

N_p = 15, N_m = 9, N_B = 5

GLTSUM =	0.	.7364810660E-14	.2760227927E-13	.5410603592E-13	.8689526314E-13	
	.1222209079E-12	.1616724472E-12	.2023691731E-12	.2463832908E-12	.2911048388E-12	
	.3387600837E-12	.3868287763E-12	.4375912716E-12	.4885615835E-12	.5420431746E-12	
	.5955658022E-12	.6514489344E-12	.7072310686E-12	.7652389519E-12	.8230223700E-12	
	.8829146199E-12	.9424724656E-12	.1004035696E-11	.1065167151E-11	.1128212316E-11	
	.1190739387E-11	.1255098882E-11	.1318863740E-11	.1384388932E-11	.1449251602E-11	
	.1515810673E-11	.1581647013E-11	.1649123050E-11	.1715822958E-11	.1784112271E-11	
	.1851578094E-11	.1920588718E-11	.1988733846E-11	.2058384215E-11	.2127131833E-11	
	.2197349603E-11	.2266631582E-11	.2337352593E-11	.2407108506E-11	.2478275850E-11	
	.2548452106E-11	.2620015312E-11	.2690564378E-11	.2762478681E-11	.2833358400E-11	.2905584098E-11
P* SUM =	0.	.3857861493E-38	.6173104715E-37	.3125045442E-36	.9876546472E-36	
	.2411245669E-35	.5000072704E-35	.9263176755E-35	.1580247434E-34	.2531238844E-34	
	.3858058857E-34	.5648557702E-34	.8000002632E-34	.1101887181E-33	.1482108279E-33	
	.1953131193E-33	.2528395892E-33	.3222253779E-33	.4050020513E-33	.5027829288E-33	
	.6172841529E-33	.7503112225E-33	.9037692305E-33	.1079635842E-32	.1280000419E-32	
	.1507038818E-32	.1763031049E-32	.2050316407E-32	.2371358798E-32	.2728704440E-32	
	.3125009900E-32	.3562972882E-32	.4045433416E-32	.4575308078E-32	.5155631890E-32	
	.5789457237E-32	.6480002106E-32	.7230553431E-32	.8044526835E-32	.8925325153E-32	
	.9876546413E-32	.1090184791E-31	.1200502826E-31	.1318983602E-31	.1446025159E-31	
	.1582030263E-31	.1727417341E-31	.1882595294E-31	.2048000661E-31	.2224073243E-31	.2411270322E-31
Q+P* SUM =	0.	.7364810660E-14	.2760227927E-13	.5410603592E-13	.8689526314E-13	
	.1222209079E-12	.1616724472E-12	.2023691731E-12	.2463832908E-12	.2911048388E-12	
	.3387600837E-12	.3868287763E-12	.4375912716E-12	.4885615835E-12	.5420431746E-12	
	.5955658022E-12	.6514489344E-12	.7072310686E-12	.7652389519E-12	.8230223700E-12	
	.8829146199E-12	.9424724656E-12	.1004035696E-11	.1065167151E-11	.1128212316E-11	
	.1190739387E-11	.1255098882E-11	.1318863740E-11	.1384388932E-11	.1449251602E-11	
	.1515810673E-11	.1581647013E-11	.1649123050E-11	.1715822958E-11	.1784112271E-11	
	.1851578094E-11	.1920588718E-11	.1988733846E-11	.2058384215E-11	.2127131833E-11	
	.2197349603E-11	.2266631582E-11	.2337352593E-11	.2407108506E-11	.2478275850E-11	
	.2548452106E-11	.2620015312E-11	.2690564378E-11	.2762478681E-11	.2833358400E-11	.2905584098E-11

Table A2-12
Form 1, Recovery Distribution Averaged, FTMP, permanent failures

T_{max} = 30 secs

N_o = 15, N_m = 9, N_B = 5

QLTSUM	= 0.	.6662993803E-17	.2658657069E-16	.5951080930E-16	.1053132541E-15	
	.1637434175E-15	.2346876172E-15	.3179035639E-15	.4132852726E-15	.5205977132E-15	
	.6397417749E-15	.7704889633E-15	.9127464042E-15	.1066291574E-14	.1231037333E-14	
	.1406766679E-14	.1593397804E-14	.1790718866E-14	.1998653057E-14	.2216993395E-14	
	.2445667802E-14	.2684473900E-14	.2933344106E-14	.3192080432E-14	.3460619584E-14	
	.3738767774E-14	.4026465819E-14	.4323523960E-14	.4629886962E-14	.4945368941E-14	
	.5269918465E-14	.5603353380E-14	.5945625919E-14	.6296557528E-14	.6656103980E-14	
	.7024090195E-14	.7400475364E-14	.7785087768E-14	.8177889899E-14	.8578713286E-14	
	.8987523618E-14	.9404155566E-14	.9828577908E-14	.1026062836E-13	.1070027868E-13	
	.1114736954E-13	.1160187560E-13	.1206364036E-13	.1253264129E-13	.1300872466E-13	.1349187064E-13
P* SUM	= 0.	.1957168207E-44	.3121476418E-43	.1581932360E-42	.4994362269E-42	
	.1220106658E-41	.2528395898E-41	.4686305537E-41	.7990979630E-41	.1280455232E-40	
	.1950922761E-40	.2857176378E-40	.4045433437E-40	.5573401105E-40	.7494664879E-40	
	.9878651959E-40	.1278556741E-39	.1629736692E-39	.2048000678E-39	.2542889940E-39	
	.3121476418E-39	.3794751826E-39	.4570153623E-39	.5460240776E-39	.6472693500E-39	
	.7621766764E-39	.8915248796E-39	.1036923130E-38	.1199146381E-38	.1380002745E-38	
	.1580247436E-38	.1801903981E-38	.2045690785E-38	.2313864453E-38	.2607088319E-38	
	.2927870930E-38	.3276801084E-38	.3656659334E-38	.4067939282E-38	.4513714744E-38	
	.4994352268E-38	.5513271391E-38	.6070678551E-38	.6670312661E-38	.7312245796E-38	
	.8000571096E-38	.8735170821E-38	.9520528389E-38	.1035630960E-37	.1124741539E-37	.1219326725E-37
Q+P* SUM	= 0.	.6662993803E-17	.2658657069E-16	.5951080930E-16	.1053132541E-15	
	.1637434175E-15	.2346876172E-15	.3179035639E-15	.4132852726E-15	.5205977132E-15	
	.6397417749E-15	.7704889633E-15	.9127464042E-15	.1066291574E-14	.1231037333E-14	
	.1406766679E-14	.1593397804E-14	.1790718866E-14	.1998653057E-14	.2216993395E-14	
	.2445667802E-14	.2684473900E-14	.2933344106E-14	.3192080432E-14	.3460619584E-14	
	.3738767774E-14	.4026465819E-14	.4323523960E-14	.4629886962E-14	.4945368941E-14	
	.5269918465E-14	.5603353380E-14	.5945625919E-14	.6296557528E-14	.6656103980E-14	
	.7024090195E-14	.7400475364E-14	.7785087768E-14	.8177889899E-14	.8578713286E-14	
	.8987523618E-14	.9404155566E-14	.9828577908E-14	.1026062836E-13	.1070027868E-13	
	.1114736954E-13	.1160187560E-13	.1206364036E-13	.1253264129E-13	.1300872466E-13	.1349187064E-13

Table A2-13

Form 1, Recovery Distribution Averaged, FTMP, permanent failures

T_{max} = 800 msecs

N_p = 15, N_m = 9, N_B = 5

Q+P* SUM =	0.	.7320683195E-14	.2730551576E-13	.5284975555E-13	.8368937348E-13	
	.1157023254E-12	.1503968806E-12	.1847078853E-12	.2207706393E-12	.2558967082E-12	
	.2924464489E-12	.3278622529E-12	.3645846547E-12	.4001033538E-12	.4368870724E-12	
	.4724423128E-12	.5092478057E-12	.5448160229E-12	.5816292473E-12	.6172020724E-12	
	.6540180413E-12	.6895925025E-12	.7264094446E-12	.7619844862E-12	.7988017725E-12	
	.8343770198E-12	.8711944271E-12	.9067697469E-12	.9435871957E-12	.9791625410E-12	
	.1015980003E-11	.1051555357E-11	.1088372822E-11	.1123948179E-11	.1160765644E-11	
	.1196341001E-11	.1233158465E-11	.1268733821E-11	.1305551283E-11	.1341126639E-11	
	.1377944099E-11	.1413519454E-11	.1450336912E-11	.1485912267E-11	.1522729722E-11	
	.1558305076E-11	.1595122530E-11	.1630697883E-11	.1667515334E-11	.1703090687E-11	.1739908136E-11

Table A2-14
FTMP, permanent failures

RM4, Recovery Rate Averaged

with $1 \bullet \frac{P_j \bar{c}_{j\ell}}{e^{-\lambda_\ell t}}$

$T_{max} = 30$ sec

$N_p = 15,\ N_m = 9,\ N_B = 5$

Q+P* SUM =	0.	.7320682969E-14	.2730551471E-13	.5284975268E-13	.8368936822E-13	
	.1157023170E-12	.1503968488E-12	.1847070697E-12	.2207706198E-12	.2558966842E-12	
	.2924464207E-12	.3278622201E-12	.3645846174E-12	.4001033118E-12	.4368870258E-12	
	.4724422612E-12	.5092477494E-12	.5448159616E-12	.5816291812E-12	.6172020012E-12	
	.6540179653E-12	.6895924213E-12	.7264093586E-12	.7619843949E-12	.7988016763E-12	
	.8343769184E-12	.8711943207E-12	.9067696353E-12	.9435870791E-12	.9791624191E-12	
	.1015979876E-11	.1051555225E-11	.1088372685E-11	.1123948036E-11	.1160765496E-11	
	.1196340848E-11	.1233158306E-11	.1268733658E-11	.1305551114E-11	.1341126465E-11	
	.1377943920E-11	.1413519270E-11	.1450336722E-11	.1485912072E-11	.1522729522E-11	
	.1558304870E-11	.1595122319E-11	.1630697667E-11	.1667515113E-11	.1703090460E-11	.1739907904E-11

Table A2-15
FTMP, permanent failures
RM4, Recovery Rate Averaged

with $\frac{P_j}{P_j^*} \bullet \frac{P_j \bar{c}_{j\ell}}{e^{-\lambda_\ell t}}$

$T_{max} = 30$ sec

$N_p = 15,\ N_m = 9,\ N_B = 5$

NOTE: Using the $\frac{P_j}{P_j^*}$ multiplier affected the 7th decimal plate.

Q+P* SUM = 0.	.7364810890E-14	.2760228028E-13	.5410603854E-13	.8689526789E-13	
.1222209154E-12	.1616724579E-12	.2023691874E-12	.2463833090E-12	.2911048614E-12	
.3387601108E-12	.3868288085E-12	.4375913090E-12	.4885616266E-12	.5420432235E-12	
.5955668573E-12	.6514489958E-12	.7072311368E-12	.7652390270E-12	.8230224523E-12	
.8829147095E-12	.9424725679E-12	.1004035801E-11	.1065167264E-11	.1128212437E-11	
.1190739517E-11	.1255099021E-11	.1318863886E-11	.1384389088E-11	.1449251744E-11	
.1515810846E-11	.1581647196E-11	.1649123243E-11	.1715823160E-11	.1784112485E-11	
.1851578315E-11	.1920588949E-11	.1988734087E-11	.2058384466E-11	.2127132094E-11	
.2197349675E-11	.2266631864E-11	.2337352885E-11	.2407108809E-11	.2478276163E-11	
.2548452430E-11	.2620015646E-11	.2690564722E-11	.2762479036E-11	.2833358766E-11	.2905584475E-11

Table A2-16 FTMP, permanent failures
RM4, Recovery Distribution Averaged with $1 \bullet \frac{P_j \bar{c}_{j\ell}}{e^{-\lambda_\ell t}}$

$T_{max} = 30$ sec

$N_p = 15$, $N_m = 9$, $N_B = 5$

Q+P* SUM = 0.	.7364810661E-14	.2760227921E-13	.5410603555E-13	.8689526227E-13	
.1222209062E-12	.1616724446E-12	.2023691691E-12	.2463832853E-12	.2911048315E-12	
.3387600745E-12	.3868287650E-12	.4375912581E-12	.4885615674E-12	.5420431560E-12	
.5955667807E-12	.6514489100E-12	.7072310410E-12	.7652389212E-12	.8230223357E-12	
.8829145822E-12	.9424724241E-12	.1004035650E-11	.1065167102E-11	.1128212262E-11	
.1190739330E-11	.1255098821E-11	.1318863673E-11	.1384388861E-11	.1449251526E-11	
.1515810592E-11	.1581646927E-11	.1649122959E-11	.1715822862E-11	.1784112170E-11	
.1851577987E-11	.1920588606E-11	.1988733729E-11	.2058384091E-11	.2127131704E-11	
.2197349468E-11	.2266631441E-11	.2337352446E-11	.2407108353E-11	.2478275691E-11	
.2548451940E-11	.2620015140E-11	.2690564199E-11	.2762478496E-11	.2833358207E-11	.2905583899E-11

Table A2-17 FTMP, permanent failures
RM4, Recovery Distribution Averaged with $\frac{P_j}{P_j^*} \bullet \frac{P_j \bar{c}_{j\ell}}{e^{-\lambda_\ell t}}$

$T_{max} = 30$ sec

$N_p = 15$, $N_m = 9$, $N_B = 5$

NOTE: Using the $\frac{P_j}{P_j^*}$ multiplier affected the 7th decimal place.

QLTSUM =	0.	.1001674270E-09	.3359771587E-09	.5117801192E-09	.7518618837E-09	
	.9300672374E-09	.1172420556E-08	.1352842874E-08	.1597332585E-08	.1779841874E-08	
	.2026341893E-08	.2210815826E-08	.2459207480E-08	.2645530920E-08	.2895702507E-08	
	.3083767082E-08	.3335613467E-08	.3525317172E-08	.3778739396E-08	.3969986223E-08	
	.4224891143E-08	.4417590722E-08	.4673890670E-08	.4867957944E-08	.5125570401E-08	
	.5320925310E-08	.5579772608E-08	.5776339797E-08	.6036348831E-08	.6234057374E-08	
	.6495159336E-08	.6693942473E-08	.6956072599E-08	.7155867493E-08	.7418964824E-08	
	.7619712332E-08	.7883719490E-08	.8085363944E-08	.8350226925E-08	.8552715926E-08	
	.8818383900E-08	.9021668131E-08	.9288093253E-08	.9492126293E-08	.9759263531E-08	
	.9964001688E-08	.1023180866E-07	.1043721080E-07	.1070564761E-07	.1091167504E-07	.1118070412E-07

P* SUM =	0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16

Q+P* SUM =	0.	.1001674270E-09	.3359771587E-09	.5117801192E-09	.7518618837E-09	
	.9300672375E-09	.1172420556E-08	.1352842874E-08	.1597332585E-08	.1779841874E-08	
	.2026341893E-08	.2210815827E-08	.2459207480E-08	.2645530920E-08	.2895702508E-08	
	.3083767083E-08	.3335613468E-08	.3525317174E-08	.3778739398E-08	.3969986225E-08	
	.4224891146E-08	.4417590726E-08	.4673890674E-08	.4867957948E-08	.5125570406E-08	
	.5320925316E-08	.5579772614E-08	.5776339805E-08	.6036348839E-08	.6234057383E-08	
	.6495159346E-08	.6693942484E-08	.6956072611E-08	.7155867507E-08	.7418964838E-08	
	.7619712348E-08	.7883719507E-08	.8085363962E-08	.8350226945E-08	.8552715948E-08	
	.8818383924E-08	.9021668157E-08	.9288093280E-08	.9492126323E-08	.9759263562E-08	
	.9964001722E-08	.1023180869E-07	.1043721084E-07	.1070564765E-07	.1091167508E-07	.1118070417E-07

Table A2-18
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min N_p = 15, N_m = 9, N_B = 5

α = 10.0

β = 1.0

QLTSUM =	0.	.1031503124E-09	.3483845773E-09	.5408108044E-09	.8011888270E-09	
	.1003584682E-08	.1271301624E-08	.1479143589E-08	.1750880627E-08	.1961711709E-08	
	.2235649334E-08	.2448122086E-08	.2723262445E-08	.2936637900E-08	.3212433750E-08	
	.3426306642E-08	.3702457864E-08	.3916605934E-08	.4192947937E-08	.4407249293E-08	
	.4683691810E-08	.4898079603E-08	.5174573130E-08	.5389010697E-08	.5665528087E-08	
	.5879995319E-08	.6156521694E-08	.6371007553E-08	.6647534729E-08	.6862033186E-08	
	.7138556695E-08	.7353064423E-08	.7629581610E-08	.7844096987E-08	.8120606909E-08	
	.8335128536E-08	.8611630254E-08	.8826157782E-08	.9102650893E-08	.9317184021E-08	
	.9593668305E-08	.9808206867E-08	.1008468220E-07	.1029922611E-07	.1057569243E-07	
	.1079024162E-07	.1106669890E-07	.1128125335E-07	.1155770156E-07	.1177226126E-07	.1204870039E-07
F* SUM =	0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999965003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+F* SUM =	0.	.1031503124E-09	.3483845773E-09	.5408108044E-09	.8011888271E-09	
	.1003584683E-08	.1271301624E-08	.1479143589E-08	.1750880627E-08	.1961711709E-08	
	.2235649334E-08	.2448122086E-08	.2723262446E-08	.2936637900E-08	.3212433751E-08	
	.3426306643E-08	.3702457866E-08	.3916605936E-08	.4192947939E-08	.4407249296E-08	
	.4683691813E-08	.4898079606E-08	.5174573134E-08	.5389010701E-08	.5665528092E-08	
	.5879995325E-08	.6156521691E-08	.6371007561E-08	.6647534737E-08	.6862033195E-08	
	.7138556705E-08	.7353064434E-08	.7629581823E-08	.7844097001E-08	.8120606923E-08	
	.8335128552E-08	.8611630271E-08	.8826157801E-08	.9102650913E-08	.9317184043E-08	
	.9593668328E-08	.9808206892E-08	.1008468223E-07	.1029922614E-07	.1057569246E-07	
	.1079024166E-07	.1106669893E-07	.1128125339E-07	.1155770160E-07	.1177226131E-07	.1204870044E-07

Table A2-19
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 10.0$

$\beta = 10.0$

QLTSUM	= 0.	.1110274791E-09	.3744608163E-09	.5774215360E-09	.8467181196E-09	
	.1050177454E-08	.1319507590E-08	.1522976457E-08	.1792297448E-08	.1995772101E-08	
	.2265083619E-08	.2468564027E-08	.2737866072E-08	.2941352233E-08	.3210644807E-08	
	.3414136719E-08	.3683419823E-08	.3886917485E-08	.4156191121E-08	.4359694530E-08	
	.4628958701E-08	.4832467856E-08	.5101722562E-08	.5305237462E-08	.5574482705E-08	
	.5778003347E-08	.6047239130E-08	.6250765513E-08	.6519991837E-08	.6723523959E-08	
	.6992740825E-08	.7196278684E-08	.7465486095E-08	.7669029690E-08	.7938227647E-08	
	.8141776976E-08	.8410965481E-08	.8614520542E-08	.8883699597E-08	.9087260388E-08	
	.9356429994E-08	.9559996514E-08	.9829156673E-08	.1003272892E-07	.1030187963E-07	
	.1050545761E-07	.1077459888E-07	.1097818257E-07	.1124731440E-07	.1145090382E-07	.1172002621E-07

P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999965003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374932272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16

Q+P* SUM	= 0.	.1110274791E-09	.3744608163E-09	.5774215360E-09	.8467181197E-09	
	.1050177454E-08	.1319507590E-08	.1522976457E-08	.1792297448E-08	.1995772101E-08	
	.2265083619E-08	.2468564027E-08	.2737866073E-08	.2941352234E-08	.3210644808E-08	
	.3414136720E-08	.3683419825E-08	.3886917486E-08	.4156191123E-08	.4359694533E-08	
	.4628958704E-08	.4832467860E-08	.5101722566E-08	.5305237466E-08	.5574482711E-08	
	.5778003353E-08	.6047239137E-08	.6250765520E-08	.6519991845E-08	.6723523968E-08	
	.6992740835E-08	.7196278695E-08	.7465486108E-08	.7669029704E-08	.7938227662E-08	
	.8141776992E-08	.8410965498E-08	.8614520561E-08	.8883699617E-08	.9087260410E-08	
	.9356430018E-08	.9559996539E-08	.9829156700E-08	.1003272895E-07	.1030187967E-07	
	.1050545764E-07	.1077459891E-07	.1097818261E-07	.1124731444E-07	.1145090386E-07	.1172002625E-07

Table A2-20
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

α = 10.0

β = 100.0

GLISUM	= 0.	.1094310260E-09	.3674442260E-09	.5606994109E-09	.8217973911E-09	
	.1015173125E-08	.1276265722E-08	.1469647657E-08	.1730730641E-08	.1924118618E-08	
	.2185191986E-08	.2378586004E-08	.2639649757E-08	.2833049814E-08	.3094103954E-08	
	.3287510048E-08	.3548554577E-08	.3741966705E-08	.4003001625E-08	.4196419787E-08	
	.4457445100E-08	.4650869293E-08	.4911885000E-08	.5105315223E-08	.5366321326E-08	
	.5559757577E-08	.5820754078E-08	.6014196355E-08	.6275183256E-08	.6468631558E-08	
	.6729608859E-08	.6923063184E-08	.7184030889E-08	.7377491235E-08	.7638449345E-08	
	.7831915710E-08	.8092864227E-08	.8286336609E-08	.8547275534E-08	.8740753932E-08	
	.9001663268E-08	.9195167679E-08	.9456087427E-08	.9649577851E-08	.9910488013E-08	
	.1010397445E-07	.1036488502E-07	.1055838747E-07	.1081927846E-07	.1101278691E-07	.1127366833E-07

F* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609275E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366364E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876025952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16

G+F* SUM	= 0.	.1094310260E-09	.3674442260E-09	.5606994110E-09	.8217973911E-09	
	.1015173125E-09	.1276265722E-08	.1469647657E-08	.1730730641E-08	.1924118619E-08	
	.2185191987E-08	.2378586005E-08	.2639649758E-08	.2833049815E-08	.3094103955E-08	
	.3287510049E-08	.3548554578E-08	.3741966707E-08	.4003001627E-08	.4196419790E-08	
	.4457445102E-08	.4650869296E-08	.4911885004E-08	.5105315228E-08	.5366321331E-08	
	.5559757583E-08	.5820754084E-08	.6014196363E-08	.6275183264E-08	.6468631567E-08	
	.6729608869E-08	.6923063195E-08	.7184030901E-08	.7377491248E-08	.7638449359E-08	
	.7831915726E-08	.8092864244E-08	.8286336627E-08	.8547275555E-08	.8740753954E-08	
	.9001663292E-08	.9195167705E-08	.9456087455E-08	.9649577880E-08	.9910488045E-08	
	.1010398448E-07	.1036488536E-07	.1055838751E-07	.1081927850E-07	.1101278695E-07	.1127366837E-07

Table A2-21

Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min N_p = 15, N_m = 9, N_B = 5

α = 10.0, β = 1000.0

```
QLTSUM   = 0.              .5851433384E-10  .1968976006E-09  .3036435806E-09  .4527681367E-09
          .5700408565E-09  .7295098229E-09  .8569646979E-09  .1026439237E-08  .1163743020E-08
          .1342895388E-08  .1489725743E-08  .1678238924E-08  .1834284067E-08  .2031851427E-08
          .2196809782E-08  .2403134773E-08  .2576714586E-08  .2791510222E-08  .2973429440E-08
          .3196418276E-08  .3386403933E-08  .3617317560E-08  .3815105674E-08  .4053684469E-08
          .4259019701E-08  .4505012540E-08  .4717647909E-08  .4970811891E-08  .5190508502E-08
          .5450608676E-08  .5677135454E-08  .5943944562E-08  .6177077999E-08  .6450376221E-08
          .6689900122E-08  .6969474836E-08  .7215180086E-08  .7500825629E-08  .7752509958E-08
          .8044027400E-08  .8301495158E-08  .8598692081E-08  .8861754025E-08  .9164444309E-08
          .9432917389E-08  .9740921008E-08  .1001462817E-07  .1032777099E-07  .1060654096E-07  .1092465455E-07

F* SUM   = 0.              .3703704376E-21  .2962962356E-20  .9999996219E-20  .2370369190E-19
          .4629626867E-19  .7999994523E-19  .1270369397E-18  .1896294543E-18  .2699997132E-18
          .3703699549E-18  .4929623513E-18  .6399991012E-18  .8137024800E-18  .1016294625E-17
          .1249997835E-17  .1517034219E-17  .1819626034E-17  .2159995489E-17  .2540364791E-17
          .2962956053E-17  .3429991663E-17  .3943693625E-17  .4506284241E-17  .5119985712E-17
          .5787020243E-17  .6509609875E-17  .7289977119E-17  .8130343842E-17  .9032932404E-17
          .9999955003E-17  .1103366384E-16  .1213625111E-16  .1330994875E-16  .1455697945E-16
          .1587956489E-16  .1727992752E-16  .1876028952E-16  .2032287309E-16  .2196990006E-16
          .2370359332E-16  .2552617433E-16  .2743986564E-16  .2944688942E-16  .3154946788E-16
          .3374982272E-16  .3605017705E-16  .3845275209E-16  .4095977050E-16  .4357345446E-16  .4629602616E-16

Q+F* SUM = 0.              .5851433384E-10  .1968976007E-09  .3036435806E-09  .4527681367E-09
          .5700408565E-09  .7295098230E-09  .8569646980E-09  .1026439237E-08  .1163743021E-08
          .1342895388E-08  .1489725743E-08  .1678238925E-08  .1834284068E-08  .2031851428E-08
          .2196809783E-08  .2403134724E-08  .2576714588E-08  .2791510224E-08  .2973429443E-08
          .3196418279E-08  .3386403937E-08  .3617317564E-08  .3815105679E-08  .4053684474E-08
          .4259019706E-08  .4505012547E-08  .4717647916E-08  .4970811899E-08  .5190508511E-08
          .5450608686E-08  .5677135465E-08  .5943944575E-08  .6177078012E-08  .6450376236E-08
          .6689900138E-08  .6969474854E-08  .7215180105E-08  .7500825650E-08  .7752509980E-08
          .8044027424E-08  .8301495184E-08  .8598692109E-08  .8861754054E-08  .9164444341E-08
          .9432917422E-08  .9740921044E-08  .1001462820E-07  .1032777103E-07  .1060654100E-07  .1092465460E-07
```

Table A2-22

Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad$ $N_p = 15$, $N_m = 9$, $N_B = 5$

α = 100.0

β = 1.0

GLTSUM	= 0.	.7554684818E-10	.2657318989E-09	.4591281505E-09	.7204570880E-09	
	.9742914694E-09	.1286976942E-08	.1584465482E-08	.1934244426E-08	.2263266023E-08	
	.2639837416E-08	.2991637660E-08	.3387561368E-08	.3755816990E-08	.4165718884E-08	
	.4545862363E-08	.4965860421E-08	.5354592319E-08	.5781882234E-08	.6176819046E-08	
	.6609374962E-08	.7008794801E-08	.7445153182E-08	.7847812090E-08	.8286915609E-08	
	.8691914885E-08	.9132999669E-08	.9539690032E-08	.9982204203E-08	.1039011656E-07	
	.1083366140E-07	.1124245685E-07	.1168674428E-07	.1209617796E-07	.1254099986E-07	
	.1295059485E-07	.1339610085E-07	.1380632934E-07	.1425181079E-07	.1466228042E-07	
	.1510795883E-07	.1551860288E-07	.1596442153E-07	.1637519180E-07	.1682110972E-07	
	.1723197137E-07	.1767795894E-07	.1808888681E-07	.1853492265E-07	.1894589855E-07	.1939196721E-07

I* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999965003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374952272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16

N** SUM	= 0.	.7554684818E-10	.2657318989E-09	.4591281505E-09	.7204570880E-09	
	.9742914694E-09	.1286976942E-08	.1584465482E-08	.1934244427E-08	.2263266023E-08	
	.2639837417E-08	.2991637660E-08	.3387561369E-08	.3755816991E-08	.4165718885E-08	
	.4545862364E-08	.4965860423E-08	.5354592320E-08	.5781882236E-08	.6176819049E-08	
	.6609374965E-08	.7008794805E-08	.7445153186E-08	.7847812094E-08	.8286915614E-08	
	.8691914891E-08	.9132999676E-08	.9539690039E-08	.9982204211E-08	.1039011657E-07	
	.1083366141E-07	.1124245686E-07	.1168674430E-07	.1209617797E-07	.1254099988E-07	
	.1295059487E-07	.1339610087E-07	.1380632935E-07	.1425181081E-07	.1466228044E-07	
	.1510795885E-07	.1551860291E-07	.1596442156E-07	.1637519183E-07	.1682110975E-07	
	.1723197146E-07	.1767795898E-07	.1808888685E-07	.1853492269E-07	.1894589859E-07	.1939196725E-07

Table A2-23
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 100.0$

$\beta = 10.0$

GLTSUM = 0.		.1275383182E-09	.4448422813E-09	.7418683990E-09	.1102530551E-08	
	.1411696350E-08	.1775746569E-08	.2085866869E-08	.2450174354E-08	.2760372772E-08	
	.3124692203E-08	.3434900066E-08	.3799212219E-08	.4109424144E-08	.4473727515E-08	
	.4783943079E-08	.5148237552E-08	.5458456721E-08	.5822742287E-08	.6132965056E-08	
	.6497241717E-08	.6807468084E-08	.7171735841E-08	.7481965806E-08	.7846224660E-08	
	.8156458221E-08	.8520708174E-08	.8830945329E-08	.9195186382E-08	.9505427130E-08	
	.9869659285E-08	.1017990362E-07	.1054412688E-07	.1085437481E-07	.1121858917E-07	
	.1152884069E-07	.1189304616E-07	.1220330127E-07	.1256749784E-07	.1287775654E-07	
	.1324194422E-07	.1355220650E-07	.1391638529E-07	.1422665115E-07	.1459082106E-07	
	.1490109050E-07	.1526525152E-07	.1557552454E-07	.1593967667E-07	.1624995327E-07	.1661409652E-07
P* SUM = 0.		.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819526034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999965003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+F* SUM = 0.		.1275383182E-09	.4448422813E-09	.7418683990E-09	.1102530551E-08	
	.1411696350E-08	.1775746569E-08	.2085866869E-08	.2450174354E-08	.2760372773E-08	
	.3124692203E-08	.3434900066E-08	.3799212220E-08	.4109424145E-08	.4473727516E-08	
	.4783943081E-08	.5148237554E-08	.5458456722E-08	.5822742289E-08	.6132965058E-08	
	.6497241720E-08	.6807468088E-08	.7171735845E-08	.7481965810E-08	.7846224665E-08	
	.8156458227E-08	.8520708180E-08	.8830945336E-08	.9195186390E-08	.9505427139E-08	
	.9869659295E-08	.1017990364E-07	.1054412689E-07	.1085437483E-07	.1121858919E-07	
	.1152884071E-07	.1189304618E-07	.1220330129E-07	.1256749786E-07	.1287775656E-07	
	.1324194424E-07	.1355220652E-07	.1391638532E-07	.1422665118E-07	.1459082109E-07	
	.1490109053E-07	.1526525155E-07	.1557552458E-07	.1593967671E-07	.1624995332E-07	.1661409657E-07

Table A2-24

Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad$ $N_{\bar{p}}$ = 15, N_m = 9, N_B = 5

α = 100.0

β = 100.0

QLTSUM	= 0.	.1167518997E-09	.3930002659E-09	.6031654004E-09	.8840232148E-09	
	.1094423810E-08	.1375282918E-08	.1585690294E-08	.1866539351E-08	.2076952943E-08	
	.2357791921E-08	.2568211726E-08	.2849040628E-08	.3059466643E-08	.3340285471E-08	
	.3550717695E-08	.3831526450E-08	.4041964882E-08	.4322763566E-08	.4533208203E-08	
	.4813996819E-08	.5024447659E-08	.5305226208E-08	.5515683249E-08	.5796451733E-08	
	.6006914974E-08	.6287673395E-08	.6498142834E-08	.6778891193E-08	.6989366829E-08	
	.7270105128E-08	.7480586958E-08	.7761315200E-08	.7971803222E-08	.8252521408E-08	
	.8463015620E-08	.8743723753E-08	.8954224154E-08	.9234922234E-08	.9445428822E-08	
	.9726116852E-08	.9936629625E-08	.1021730761E-07	.1042782656E-07	.1070849450E-07	
	.1091901963E-07	.1119967753E-07	.1141020884E-07	.1169085669E-07	.1190139418E-07	.1218203199E-07

P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955063E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16

Q+P* SUM	= 0.	.1167518997E-09	.3930002659E-09	.6031654004E-09	.8840232149E-09	
	.1094423810E-08	.1375282918E-08	.1585690294E-08	.1866539351E-08	.2076952943E-08	
	.2357791921E-08	.2568211726E-08	.2849040628E-08	.3059466644E-08	.3340285472E-08	
	.3550717696E-08	.3831526452E-08	.4041964883E-08	.4322763568E-08	.4533208205E-08	
	.4813996822E-08	.5024447662E-08	.5305226212E-08	.5515683254E-08	.5796451738E-08	
	.6006914980E-08	.6287673401E-08	.6498142841E-08	.6778891202E-08	.6989366838E-08	
	.72701051[illegible]8E-08	.7480586969E-08	.7761315212E-08	.7971803235E-08	.8252521423E-08	
	.8463015636E-08	.8743723770E-08	.8954224173E-08	.9234922255E-08	.9445428844E-08	
	.9726116876E-08	.9936629650E-08	.1021730763E-07	.1042782659E-07	.1070849453E-07	
	.1091901967E-07	.1119967756E-07	.1141020888E-07	.1169085673E-07	.1190139423E-07	.1218203204E-07

Table A2-25
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15,\ N_m = 9,\ N_B = 5$

α = 100.0

β 1000.0

GLTSUM	= 0.	.1384237383E-10	.4857729234E-10	.8383195959E-10	.1331513958E-09	
	.1829480998E-09	.2467640861E-09	.3110151241E-09	.3892402225E-09	.4678584197E-09	
	.5604057186E-09	.6533044252E-09	.7600875363E-09	.8671806239E-09	.9881136790E-09	
	.1109315538E-08	.1244313186E-08	.1379538722E-08	.1528516125E-08	.1677680757E-08	
	.1840553589E-08	.2003573243E-08	.2180257686E-08	.2357048796E-08	.2547461534E-08	
	.2737941038E-08	.2941999260E-08	.3146084593E-08	.3363705984E-08	.3581315081E-08	
	.3812417824E-08	.4043469112E-08	.4287971881E-08	.4532384281E-08	.4790206239E-08	
	.5047899157E-08	.5318959957E-08	.5589853286E-08	.5874073061E-08	.6158087176E-08	
	.6455336542E-08	.6752442298E-08	.7062742348E-08	.7372761076E-08	.7695983379E-08	
	.8018886884E-08	.8354953481E-08	.8690664038E-08	.9039497439E-08	.9387937793E-08	.9749460975E-08
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1384237383E-10	.4857729235E-10	.8383195960E-10	.1331513958E-09	
	.1829480999E-09	.2467640862E-09	.3110151242E-09	.3892402227E-09	.4678584200E-09	
	.5604057190E-09	.6533044257E-09	.7600875369E-09	.8671806247E-09	.9881136800E-09	
	.1109315539E-08	.1244313187E-08	.1379538724E-08	.1528516128E-08	.1677680760E-08	
	.1840553592E-08	.2003573247E-08	.2180257690E-08	.2357048801E-08	.2547461540E-08	
	.2737941044E-08	.2941999266E-08	.3146084600E-08	.3363705992E-08	.3581315090E-08	
	.3812417834E-08	.4043469123E-08	.4287971893E-08	.4532384294E-08	.4790206254E-08	
	.5047899173E-08	.5318959974E-08	.5589853305E-08	.5874073081E-08	.6158087198E-08	
	.6455336565E-08	.6752442324E-08	.7062742375E-08	.7372761106E-08	.7695983411E-08	
	.8018886918E-08	.8354953517E-08	.8690664077E-08	.9039497480E-08	.9387937836E-08	.9749461021E-08

Table A2-26
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad$ N_p = 15, N_m = 9, N_B = 5

α = 1000.0

β = 1.0

GLTSUM	= 0.	.4255585817E-10	.1633645269E-09	.3412388751E-09	.5873251904E-09	
	.8867451547E-09	.1250754554E-08	.1664587105E-08	.2139601220E-08	.2661136007E-08	
	.3240646080E-08	.3863569727E-08	.4541451758E-08	.5259823831E-08	.6030315329E-08	
	.6838545838E-08	.7696224054E-08	.8589052395E-08	.9528813937E-08	.1050128915E-07	
	.1151833088E-07	.1256579316E-07	.1365559425E-07	.1477365757E-07	.1593196274E-07	
	.1711649861E-07	.1833930232E-07	.1958642445E-07	.2086995610E-07	.2217600610E-07	
	.2351671613E-07	.2487825001E-07	.2627279683E-07	.2768657234E-07	.2913181005E-07	
	.3059477477E-07	.3208774165E-07	.3359702182E-07	.3513492946E-07	.3668781947E-07	
	.3826804258E-07	.3986199507E-07	.4148206190E-07	.4311467846E-07	.4477226177E-07	
	.4644128422E-07	.4813419281E-07	.4983749496E-07	.5156366569E-07	.5329924557E-07	.5505673586E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.4255585817E-10	.1633645269E-09	.3412388751E-09	.5873251904E-09	
	.8867451548E-09	.1250754554E-08	.1664587105E-08	.2139601220E-08	.2661136007E-08	
	.3240646080E-08	.3863569728E-08	.4541451759E-08	.5259823832E-08	.6030315330E-08	
	.6838545839E-08	.7696224056E-08	.8589052397E-08	.9528813939E-08	.1050128916E-07	
	.1151833088E-07	.1256579316E-07	.1365559425E-07	.1477365758E-07	.1593196275E-07	
	.1711649862E-07	.1833930232E-07	.1958642446E-07	.2086995611E-07	.2217600611E-07	
	.2351671614E-07	.2487825003E-07	.2627279684E-07	.2768657235E-07	.2913181007E-07	
	.3059477478E-07	.3208774167E-07	.3359702184E-07	.3513492948E-07	.3668781950E-07	
	.3826804261E-07	.3986199510E-07	.4148206192E-07	.4311467849E-07	.4477226180E-07	
	.4644128426E-07	.4813419285E-07	.4983749500E-07	.5156366573E-07	.5329924561E-07	.5505673591E-07

Table A2-27
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad N_p = 15,\ N_m = 9,\ N_B = 5$

$\alpha = 1000.0$

$\beta = 10.0$

GLTSUM	= 0.	.1682368693E-09	.6430058138E-09	.1314476427E-08	.2166866434E-08	
	.3123638679E-08	.4191629838E-08	.5311361258E-08	.6502526291E-08	.7715370127E-08	
	.8976917642E-08	.1024296950E-07	.1154473462E-07	.1284119129E-07	.1416593556E-07	
	.1547976522E-07	.1681763686E-07	.1814139176E-07	.1948676040E-07	.2081618406E-07	
	.2216583189E-07	.2349849170E-07	.2485057969E-07	.2618508537E-07	.2753856249E-07	
	.2887411949E-07	.3022838507E-07	.3156453928E-07	.3291925003E-07	.3425574193E-07	
	.3561070164E-07	.3694738291E-07	.3830247946E-07	.3963926532E-07	.4099443464E-07	
	.4233127665E-07	.4368648211E-07	.4502335258E-07	.4637857324E-07	.4771545635E-07	
	.4907058026E-07	.5040756696E-07	.5176278728E-07	.5309967241E-07	.5445488522E-07	
	.5579176582E-07	.5714696890E-07	.5848384329E-07	.5983903536E-07	.6117590256E-07	.6253108290E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1682368693E-09	.6430058138E-09	.1314476427E-08	.2166866434E-08	
	.3123638679E-08	.4191629838E-08	.5311361258E-08	.6502526291E-08	.7715370127E-08	
	.8976917643E-08	.1024296951E-07	.1154473462E-07	.1284119130E-07	.1416593556E-07	
	.1547976522E-07	.1681763686E-07	.1814139176E-07	.1948676040E-07	.2081618407E-07	
	.2216583189E-07	.2349849171E-07	.2485057969E-07	.2618508538E-07	.2753856250E-07	
	.2887411950E-07	.3022838508E-07	.3156453929E-07	.3291925003E-07	.3425574193E-07	
	.3561070165E-07	.3694738292E-07	.3830247948E-07	.3963926534E-07	.4099443466E-07	
	.4233127667E-07	.4368648213E-07	.4502335260E-07	.4637857326E-07	.4771545638E-07	
	.4907068028E-07	.5040756699E-07	.5176278730E-07	.5309967244E-07	.5445488525E-07	
	.5579176586E-07	.5714696894E-07	.5848384333E-07	.5983903540E-07	.6117590260E-07	.6253108294E-07

Table A2-28

Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 1000.0$

$\beta = 100.0$

GLTSUM =	0.	.1726971851E-09	.5981841251E-09	.9806736212E-09	.1447341333E-08	
	.1837947323E-08	.2306199795E-08	.2697125834E-08	.3165427901E-08	.3556371701E-08	
	.4024663841E-08	.4415613680E-08	.4883893590E-08	.5274849011E-08	.5743116601E-08	
	.6134077586E-08	.6602332856E-08	.6993299401E-08	.7461542352E-08	.7852514456E-08	
	.8320745090E-08	.8711722750E-08	.9179941070E-08	.9570924284E-08	.1003913029E-07	
	.1043011906E-07	.1089831276E-07	.1128930707E-07	.1175748846E-07	.1214848833E-07	
	.1261665741E-07	.1300766282E-07	.1347581960E-07	.1386683055E-07	.1433497503E-07	
	.1472599153E-07	.1519412370E-07	.1558514574E-07	.1605326562E-07	.1644429320E-07	
	.1691240078E-07	.1730343389E-07	.1777152918E-07	.1816256782E-07	.1863065082E-07	
	.1902159500E-07	.1948976570E-07	.1988081541E-07	.2034887383E-07	.2073992906E-07	.2120797520E-07
F* SUM =	0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626667E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
G+F* SUM =	0.	.1726971851E-09	.5981841251E-09	.9806736212E-09	.1447341333E-08	
	.1837947323E-08	.2306199796E-08	.2697125834E-08	.3165427901E-08	.3556371701E-08	
	.4024663842E-08	.4415613681E-08	.4883893590E-08	.5274849012E-08	.5743116603E-08	
	.6134077587E-08	.6602332857E-08	.6993299403E-08	.7461542354E-08	.7852514458E-08	
	.8320745093E-08	.8711722753E-08	.9179941074E-08	.9570924289E-08	.1003913030E-07	
	.1043011906E-07	.1089831276E-07	.1128930708E-07	.1175748847E-07	.1214848834E-07	
	.1261665742E-07	.1300766283E-07	.1347581961E-07	.1386683057E-07	.1433497505E-07	
	.1472599154E-07	.1519412372E-07	.1558514576E-07	.1605326564E-07	.1644429322E-07	
	.1691240080E-07	.1730343391E-07	.1777152921E-07	.1816256785E-07	.1863065085E-07	
	.1902159503E-07	.1948976574E-07	.1988081545E-07	.2034887387E-07	.2073992911E-07	.2120797525E-07

Table A2-29
Form 1, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 1000.0$, $\beta = 1000.0$

```
QLTSUM    = 0.               .1000879280E-09  .3356811282E-09  .5112122001E-09  .7509726847E-09
            .9288231283E-09  .1170792091E-08  .1350804922E-08  .1594853545E-08  .1776908865E-08
            .2022930015E-08  .2206918391E-08  .2454805272E-08  .2640622502E-08  .2890273519E-08
            .3077820515E-08  .3329139036E-08  .3518321603E-08  .3771215821E-08  .3961944483E-08
            .4216327218E-08  .4408517038E-08  .4664305552E-08  .4857875943E-08  .5114991762E-08
            .5309866305E-08  .5568235029E-08  .5764341296E-08  .6023892428E-08  .6221161800E-08
            .6481828566E-08  .6680196065E-08  .6941915243E-08  .7141319358E-08  .7404031116E-08
            .7604413642E-08  .7868061370E-08  .8069367250E-08  .8333897409E-08  .8536074575E-08
            .8801436543E-08  .9004435776E-08  .9270581703E-08  .9474356483E-08  .9741241152E-08
            .9945747525E-08  .1021332822E-07  .1041852466E-07  .1068676103E-07  .1089260834E-07  .1116146228E-07

P* SUM    = 0.               .3703704376E-21  .2962962356E-20  .9999996219E-20  .2370369190E-19
            .4629626867E-19  .7999994523E-19  .1270369397E-18  .1896294543E-18  .2699997132E-18
            .3703699549E-18  .4929623513E-18  .6399991012E-18  .8137024800E-18  .1016294625E-17
            .1249997835E-17  .1517034219E-17  .1819626034E-17  .2159995489E-17  .2540364791E-17
            .2962956053E-17  .3429991663E-17  .3943693625E-17  .4506284241E-17  .5119985712E-17
            .5787020243E-17  .6509609875E-17  .7289977119E-17  .8130343842E-17  .9032932404E-17
            .9999955003E-17  .1103366394E-16  .1213625111E-16  .1330994875E-16  .1455697945E-16
            .1587956489E-16  .1727992752E-16  .1876028952E-16  .2032287309E-16  .2196990006E-16
            .2370359332E-16  .2552617433E-16  .2743986564E-16  .2944688942E-16  .3154946788E-16
            .3374952272E-16  .3605017705E-16  .3845275209E-16  .4095977050E-16  .4357345446E-16  .4629602616E-16

Q+P* SUM  = 0.               .1000879280E-09  .3356811282E-09  .5112122001E-09  .7509726847E-09
            .9288231283E-09  .1170792091E-08  .1350804922E-08  .1594853545E-08  .1776908865E-08
            .2022930016E-08  .2206918392E-08  .2454805273E-08  .2640622503E-08  .2890273520E-08
            .3077820516E-08  .3329139038E-08  .3518321604E-08  .3771215823E-08  .3961944486E-08
            .4216327221E-08  .4408517041E-08  .4664305556E-08  .4857875947E-08  .5114991767E-08
            .5309866311E-08  .5568235036E-08  .5764341303E-08  .6023892436E-08  .6221161809E-08
            .6481828576E-08  .6680196076E-08  .6941915255E-08  .7141319372E-08  .7404031131E-08
            .7604413658E-08  .7868061388E-08  .8069367268E-08  .8333897429E-08  .8536074597E-08
            .8801436566E-08  .9004435801E-08  .9270581730E-08  .9474356512E-08  .9741241184E-08
            .9945747559E-08  .1021332826E-07  .1041852470E-07  .1068676108E-07  .1089260838E-07  .1116146233E-07
```

Table A2-30
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 10.0$

$\beta = 1.0$

QLTSUM	= 0.	.1025516990E-09	.3463001117E-09	.5374402016E-09	.7965956073E-09	
	.9981630531E-09	.1265087835E-08	.1472468120E-08	.1743710846E-08	.1954315158E-08	
	.2227940345E-08	.2440325298E-08	.2715258533E-08	.2928625443E-08	.3204273603E-08	
	.3418182320E-08	.3694218958E-08	.3908427453E-08	.4184673184E-08	.4399048530E-08	
	.4675404873E-08	.4889874111E-08	.5166287025E-08	.5380810123E-08	.5657249961E-08	
	.5871804960E-08	.6148255462E-08	.6362830316E-08	.6639282564E-08	.6853870665E-08	
	.7130319768E-08	.7344917492E-08	.7621360768E-08	.7835966127E-08	.8112402107E-08	
	.8327014009E-08	.8603441886E-08	.8818059732E-08	.9094479067E-08	.9309102528E-08	
	.9585513079E-08	.9800141972E-08	.1007654361E-07	.1029117783E-07	.1056757048E-07	
	.1078220999E-07	.1105859361E-07	.1127323836E-07	.1154961294E-07	.1176426291E-07	.1204062844E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1025516990E-09	.3463001117E-09	.5374402016E-09	.7965956073E-09	
	.9981630532E-09	.1265087835E-08	.1472468120E-08	.1743710846E-08	.1954315158E-08	
	.2227940346E-08	.2440325299E-08	.2715258534E-08	.2928625444E-08	.3204273604E-08	
	.3418182322E-08	.3694218959E-08	.3908427454E-08	.4184673186E-08	.4399048532E-08	
	.4675404876E-08	.4889874114E-08	.5166287029E-08	.5380810127E-08	.5657249966E-08	
	.5871804966E-08	.6148255469E-08	.6362830324E-08	.6639282572E-08	.6853870674E-08	
	.7130319778E-08	.7344917503E-08	.7621360780E-08	.7835966141E-08	.8112402121E-08	
	.8327014025E-08	.8603441904E-08	.8818059751E-08	.9094479087E-08	.9309102550E-08	
	.9585513102E-08	.9800141997E-08	.1007654363E-07	.1029117786E-07	.1056757051E-07	
	.1078221002E-07	.1105859365E-07	.1127323840E-07	.1154961298E-07	.1176426296E-07	.1204062848E-07

Table A2-31
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 10.0$

$\beta = 10.0$

```
QLTSUM   = 0.               .1103044975E-09  .3724382414E-09  .5758286812E-09  .8445645169E-09
           .1048572892E-08  .1317353616E-08  .1521372389E-08  .1790144183E-08  .1994168622E-08
           .2262931072E-08  .2466961138E-08  .2735714242E-08  .2939749935E-08  .3208493694E-08
           .3412535011E-08  .3681269427E-08  .3885316367E-08  .4154041442E-08  .4358094003E-08
           .4626809739E-08  .4830867918E-08  .5099574318E-08  .5303638114E-08  .5572335178E-08
           .5776404590E-08  .6045092320E-08  .6249167346E-08  .6517845743E-08  .6721926382E-08
           .6990595448E-08  .7194681698E-08  .7463341436E-08  .7667433294E-08  .7936083704E-08
           .8140181170E-08  .8408822255E-08  .8612925326E-08  .8881557087E-08  .9085665762E-08
           .9354288201E-08  .9558402478E-08  .9827015597E-08  .1003113547E-07  .1029973928E-07
           .1050386475E-07  .1077245923E-07  .1097659031E-07  .1124517548E-07  .1144931214E-07  .1171788800E-07

P* SUM   = 0.               .3703704376E-21  .2962962356E-20  .9999996219E-20  .2370369190E-19
           .4629626867E-19  .7999994523E-19  .1270369397E-18  .1896294543E-18  .2699997132E-18
           .3703699549E-18  .4929623513E-18  .6399991012E-18  .8137024800E-18  .1016294625E-17
           .1249997835E-17  .1517034219E-17  .1819626034E-17  .2159995489E-17  .2540364791E-17
           .2962956053E-17  .3429991663E-17  .3943693625E-17  .4506284241E-17  .5119985712E-17
           .5787020243E-17  .6509609875E-17  .7289977119E-17  .8130343842E-17  .9032932404E-17
           .9999955003E-17  .1103366384E-16  .1213625111E-16  .1330994875E-16  .1455697945E-16
           .1587956489E-16  .1727992752E-16  .1876028952E-16  .2032287309E-16  .2196990006E-16
           .2370359332E-16  .2557617433E-16  .2743986564E-16  .2944688942E-16  .3154946788E-16
           .3374982272E-16  .3605017705E-16  .3845275209E-16  .4095977050E-16  .4357345446E-16  .4629602616E-16

Q+P* SUM = 0.               .1103044975E-09  .3724382414E-09  .5758286812E-09  .8445645169E-09
           .1048572892E-08  .1317353616E-08  .1521372389E-08  .1790144183E-08  .1994168622E-08
           .2262931072E-08  .2466961139E-08  .2735714242E-08  .2939749935E-08  .3208493695E-08
           .3412535012E-08  .3681269429E-08  .3885316368E-08  .4154041444E-08  .4358094005E-08
           .4626809742E-08  .4830867922E-08  .5099574322E-08  .5303638119E-08  .5572335183E-08
           .5776404596E-08  .6045092326E-08  .6249167353E-08  .6517845751E-08  .6721926391E-08
           .6990595458E-08  .7194681709E-08  .7463341448E-08  .7667433307E-08  .7936083719E-08
           .8140181186E-08  .8408822272E-08  .8612925344E-08  .8881557108E-08  .9085665784E-08
           .9354288225E-08  .9558402504E-08  .9827015625E-08  .1003113550E-07  .1029973931E-07
           .1050386478E-07  .1077245927E-07  .1097659035E-07  .1124517552E-07  .1144931219E-07  .1171788805E-07
```

Table A2-32

Form 2, FTMP, intermittent, no restrictions.

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 10.0$

$\beta = 100.0$

QLTSUM =	0.	.1094042144E-09	.3673720002E-09	.5606517687E-09	.8217243359E-09	
	.1015125460E-08	.1276192680E-08	.1469600001E-08	.1730657614E-08	.1924070971E-08	
	.2185118974E-08	.2378538365E-08	.2639576759E-08	.2833002184E-08	.3094030971E-08	
	.3287452426E-08	.3548481608E-08	.3741919093E-08	.4002928671E-08	.4196372183E-08	
	.4457372159E-08	.4650821698E-08	.4911812074E-08	.5105267636E-08	.5366248415E-08	
	.5559709999E-08	.5820681181E-08	.6014148786E-08	.6275110374E-08	.6468583997E-08	
	.6729535992E-08	.6923015632E-08	.7183958036E-08	.7377443691E-08	.7638376507E-08	
	.7831868175E-08	.8092791403E-08	.8286289083E-08	.8547202725E-08	.8740706415E-08	
	.9001610473E-08	.9195120171E-08	.9456014647E-08	.9649530351E-08	.9910415247E-08	
	.1010393696E-07	.1036481227E-07	.1055833998E-07	.1081920573E-07	.1101273944E-07	.1127359560E-07
P* SUM =	0.	.3703704438E-21	.2962962455E-20	.9999996719E-20	.2370369348E-19	
	.4629627253E-19	.7999995323E-19	.1270369545E-18	.1896294796E-18	.2699997537E-18	
	.3703700166E-18	.4929624416E-18	.6399992292E-18	.8137026563E-18	.1016294862E-17	
	.1249998147E-17	.1517034624E-17	.1819626550E-17	.2159996137E-17	.2540365595E-17	
	.2962957041E-17	.3429992863E-17	.3943695071E-17	.4506285968E-17	.5119987760E-17	
	.5787022654E-17	.6509612696E-17	.7289980400E-17	.8130347636E-17	.9032936770E-17	
	.9999970003E-17	.1103366954E-16	.1213625758E-16	.1330995607E-16	.1455698770E-16	
	.1587957416E-16	.1727993789E-16	.1876030109E-16	.2032288596E-16	.2196991434E-16	
	.2370350913E-16	.2552619178E-16	.2743988485E-16	.2944691053E-16	.3154949102E-16	
	.3374984803E-16	.3605620469E-16	.3845278221E-16	.4095980327E-16	.4357349005E-16	.4629606474E-16
Q+P* SUM =	0.	.1094042144E-09	.3673720002E-09	.5606517687E-09	.8217243359E-09	
	.1015125460E-08	.1276192680E-08	.1469600001E-08	.1730657614E-08	.1924070971E-08	
	.2185118974E-08	.2378538366E-08	.2639576760E-08	.2833002185E-08	.3094030972E-08	
	.3287452427E-08	.3548481609E-08	.3741919094E-08	.4002928673E-08	.4196372186E-08	
	.4457372162E-08	.4650821701E-08	.4911812078E-08	.5105267641E-08	.5366248420E-08	
	.5559710005E-08	.5820681188E-08	.6014148793E-08	.6275110382E-08	.6468584006E-08	
	.6729536002E-08	.6923015643E-08	.7183958048E-08	.7377443705E-08	.7638376521E-08	
	.7831868191E-08	.8092791420E-08	.8286289101E-08	.8547202745E-08	.8740706437E-08	
	.9001610497E-08	.9195120196E-08	.9456014675E-08	.9649530380E-08	.9910415279E-08	
	.1010393699E-07	.1036481231E-07	.1055834002E-07	.1081920577E-07	.1101273948E-07	.1127359565E-07

Table A2-33
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

α = 10.0

β = 1000.0

```
QLTSUM    = 0.                .5763538543E-10  .1933877387E-09  .2958153655E-09  .4390972411E-09
            .5490483477E-09  .6998583807E-09  .8173547760E-09  .9757055631E-09  .1100751045E-08
            .1266638018E-08  .1399220058E-08  .1572623126E-08  .1712714450E-08  .1893599523E-08
            .2041159574E-08  .2229479971E-08  .2384456260E-08  .2580154001E-08  .2742483356E-08
            .2945490368E-08  .3115100086E-08  .3325339304E-08  .3502148209E-08  .3719534598E-08
            .3903454025E-08  .4127895517E-08  .4318830211E-08  .4550228574E-08  .4748077517E-08
            .4986329144E-08  .5190986321E-08  .5435982966E-08  .5647338064E-08  .5898967511E-08
            .6116906561E-08  .6375053240E-08  .6599459208E-08  .6864004758E-08  .7094758082E-08
            .7365581869E-08  .7602560954E-08  .7879540541E-08  .8122622211E-08  .8405633789E-08
            .8654693697E-08  .8943612477E-08  .9198525479E-08  .9493226350E-08  .9753866547E-08  .1005422320E-07

P* SUM    = 0.                .3703704376E-21  .2962962356E-20  .9999996219E-20  .2370369190E-19
            .4629626867E-19  .7999994523E-19  .1270369397E-18  .1896294543E-18  .2699997132E-18
            .3703699549E-18  .4929623513E-18  .6399991012E-18  .8137024800E-18  .1016294625E-17
            .1249997835E-17  .1517034219E-17  .1819626034E-17  .2159995489E-17  .2540364791E-17
            .2962956053E-17  .3429991663E-17  .3943693625E-17  .4506284241E-17  .5119985712E-17
            .5787020243E-17  .6509609875E-17  .7289977119E-17  .8130343842E-17  .9032932404E-17
            .9999955003E-17  .1103366384E-16  .1213625111E-16  .1330994875E-16  .1455697945E-16
            .1587956489E-16  .1727992752E-16  .1876028952E-16  .2032287309E-16  .2196990006E-16
            .2370359332E-16  .2552617433E-16  .2743986564E-16  .2944688942E-16  .3154946788E-16
            .3374982272E-16  .3605017705E-16  .3845275209E-16  .4095977050E-16  .4357345446E-16  .4629602616E-16

Q+P* SUM  = 0.                .5763538543E-10  .1933877387E-09  .2958153655E-09  .4390972411E-09
            .5490483477E-09  .6998583808E-09  .8173547761E-09  .9757055633E-09  .1100751046E-08
            .1266638018E-08  .1399220058E-08  .1572623126E-08  .1712714451E-08  .1893599524E-08
            .2041159575E-08  .2229479972E-08  .2384456261E-08  .2580154003E-08  .2742483359E-08
            .2945490371E-08  .3115100089E-08  .3325339308E-08  .3502148213E-08  .3719534603E-08
            .3903454031E-08  .4127895524E-08  .4318830219E-08  .4550228582E-08  .4748077526E-08
            .4986329154E-08  .5190986332E-08  .5435982978E-08  .5647338077E-08  .5898967526E-08
            .6116906577E-08  .6375053258E-08  .6599459226E-08  .6864004779E-08  .7094758104E-08
            .7365581893E-08  .7602560980E-08  .7879540569E-08  .8122622241E-08  .8405633820E-08
            .8654693731E-08  .8943612513E-08  .9198525518E-08  .9493226391E-08  .9753866590E-08  .1005422325E-07
```

Table A2-34
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad$ $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 100.0$

$\beta = 1.0$

GLTSUM	= 0.	.6883737055E-10	.2405349475E-09	.4100723914E-09	.6442586259E-09	
	.8712299551E-09	.1157271615E-08	.1430506817E-08	.1757425750E-08	.2066531202E-08	
	.2424732289E-08	.2761007566E-08	.3142705509E-08	.3499253887E-08	.3898386530E-08	
	.4269912002E-08	.4681875218E-08	.5064390856E-08	.5485743183E-08	.5876286094E-08	
	.6304482112E-08	.6700869379E-08	.7134039784E-08	.7534672595E-08	.7971451306E-08	
	.8375163410E-08	.8814555478E-08	.9220498560E-08	.9661781052E-08	.1006933928E-07	
	.1051198778E-07	.1092071472E-07	.1136434927E-07	.1177392158E-07	.1221826710E-07	
	.1262845080E-07	.1307330825E-07	.1348393407E-07	.1392915951E-07	.1434010504E-07	
	.1478559441E-07	.1519677115E-07	.1564244920E-07	.1605379320E-07	.1649960556E-07	
	.1691107058E-07	.1735697795E-07	.1776853059E-07	.1821450457E-07	.1862612069E-07	.1907214076E-07

P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16

G+P* SUM	= 0.	.6883737055E-10	.2405349475E-09	.4100723914E-09	.6442586259E-09	
	.8712299551E-09	.1157271615E-08	.1430506817E-08	.1757425750E-08	.2066531202E-08	
	.2424732290E-08	.2761007567E-08	.3142705510E-08	.3499253887E-08	.3898386531E-08	
	.4269912003E-08	.4681875220E-08	.5064390858E-08	.5485743185E-08	.5876286096E-08	
	.6304482115E-08	.6700869382E-08	.7134039788E-08	.7534672599E-08	.7971451311E-08	
	.8375163415E-08	.8814555484E-08	.9220498567E-08	.9661781060E-08	.1006933929E-07	
	.1051198779E-07	.1092071473E-07	.1136434928E-07	.1177392160E-07	.1221826711E-07	
	.1262845081E-07	.1307330827E-07	.1348393408E-07	.1392915954E-07	.1434010506E-07	
	.1478559443E-07	.1519677117E-07	.1564244923E-07	.1605379323E-07	.1649960559E-07	
	.1691107061E-07	.1735697799E-07	.1776853063E-07	.1821450462E-07	.1862612073E-07	.1907214081E-07

Table A2-35
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad$ $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 100.0$

$\beta = 10.0$

QLTSUM	= 0.	.1193450737E-09	.4209101281E-09	.7192733657E-09	.1073944768E-08	
	.1387784091E-08	.1746798815E-08	.2061859253E-08	.2421207365E-08	.2736365763E-08	
	.3095732843E-08	.3410901184E-08	.3770262592E-08	.4085433978E-08	.4444787787E-08	
	.4759961676E-08	.5119307734E-08	.5434484083E-08	.5793822380E-08	.6109001184E-08	
	.6468331721E-08	.6783512979E-08	.7142835757E-08	.7458019467E-08	.7817334486E-08	
	.8132520648E-08	.8491827910E-08	.8807016522E-08	.9166316028E-08	.9481507090E-08	
	.9840798841E-08	.1015599235E-07	.1051527635E-07	.1083047230E-07	.1118974855E-07	
	.1150494695E-07	.1186421545E-07	.1217941629E-07	.1253867704E-07	.1285388033E-07	
	.1321313332E-07	.1352833905E-07	.1388758430E-07	.1420279247E-07	.1456202997E-07	
	.1487724059E-07	.1523647034E-07	.1555168339E-07	.1591090540E-07	.1622612089E-07	.1658533516E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819526034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1193450737E-09	.4209101281E-09	.7192733657E-09	.1073944768E-08	
	.1387784091E-08	.1746798815E-08	.2061859253E-08	.2421207365E-08	.2736365764E-08	
	.3095732843E-08	.3410901184E-08	.3770262593E-08	.4085433979E-08	.4444787788E-08	
	.4759961677E-08	.5119307736E-08	.5434484085E-08	.5793822382E-08	.6109001187E-08	
	.6468331724E-08	.6783512982E-08	.7142835760E-08	.7458019471E-08	.7817334491E-08	
	.8132520654E-08	.8491827917E-08	.8807016529E-08	.9166316036E-08	.9481507099E-08	
	.9840798851E-08	.1015599236E-07	.1051527636E-07	.1083047232E-07	.1118974856E-07	
	.1150494697E-07	.1186421546E-07	.1217941631E-07	.1253867706E-07	.1285388035E-07	
	.1321313334E-07	.1352833908E-07	.1388758433E-07	.1420279250E-07	.1456203001E-07	
	.1487724062E-07	.1523647038E-07	.1555168343E-07	.1591090545E-07	.1622612094E-07	.1658533521E-07

Table A2-36

Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 100.0$

$\beta = 100.0$

QLTSUM =	0.	.1164755532E-09	.3922534338E-09	.6026645300E-09	.8832645600E-09	
	.1093922437E-08	.1374524385E-08	.1585189011E-08	.1865780968E-08	.2076451751E-08	
	.2357033690E-08	.2567710626E-08	.2848282547E-08	.3058965635E-08	.3339527541E-08	
	.3550216778E-08	.3830768672E-08	.4041464056E-08	.4322005939E-08	.4532707469E-08	
	.4813239342E-08	.5023947017E-08	.5304468882E-08	.5515182699E-08	.5795694559E-08	
	.6006414515E-08	.6286916371E-08	.6497642467E-08	.6778134321E-08	.6988866553E-08	
	.7269348407E-08	.7480086774E-08	.7760558629E-08	.7971303129E-08	.8251764989E-08	
	.8462515619E-08	.8742967484E-08	.8953724244E-08	.9234166116E-08	.9444929004E-08	
	.9725360885E-08	.9936129899E-08	.1021655179E-07	.1042732693E-07	.1070773883E-07	
	.1091852009E-07	.1119892201E-07	.1140970939E-07	.1169010133E-07	.1190089482E-07	.1218127678E-07

F* SUM =	0.	.3703704438E-21	.2962962455E-20	.9999996719E-20	.2370369348E-19	
	.4629627253E-19	.7999995323E-19	.1270369545E-18	.1896294796E-18	.2699997537E-18	
	.3703700166E-18	.4929624416E-18	.6399992292E-18	.8137026563E-18	.1016294862E-17	
	.1249998147E-17	.1517034624E-17	.1819626550E-17	.2159996137E-17	.2540365595E-17	
	.2962957041E-17	.3429992863E-17	.3943695071E-17	.4506285968E-17	.5119987760E-17	
	.5787022654E-17	.6509612696E-17	.7289980400E-17	.8130347636E-17	.9032936770E-17	
	.9999970003E-17	.1103366954E-16	.1213625758E-16	.1330995607E-16	.1455698770E-16	
	.1587957416E-16	.1727993789E-16	.1876030109E-16	.2032288596E-16	.2196991434E-16	
	.2370360913E-16	.2552619178E-16	.2743988485E-16	.2944691053E-16	.3154949102E-16	
	.3374954803E-16	.3605020469E-16	.3845278221E-16	.4095980327E-16	.4357349005E-16	.4629606474E-16

Q+P* SUM =	0.	.1164755532E-09	.3922534338E-09	.6026645300E-09	.8832645600E-09	
	.1093922437E-08	.1374524385E-08	.1585189011E-08	.1865780969E-08	.2076451751E-08	
	.2357033690E-08	.2567710626E-08	.2848282548E-08	.3058965635E-08	.3339527542E-08	
	.3550216779E-08	.3830768673E-08	.4041464058E-08	.4322005941E-08	.4532707472E-08	
	.4813239345E-08	.5023947020E-08	.5304468886E-08	.5515182703E-08	.5795694564E-08	
	.6006414521E-08	.6286916378E-08	.6497642474E-08	.6778134329E-08	.6988866562E-08	
	.7269348417E-08	.7480086785E-08	.7760558642E-08	.7971303143E-08	.8251765003E-08	
	.8462515635E-08	.8742967501E-08	.8953724263E-08	.9234166137E-08	.9444929026E-08	
	.9725360909E-08	.9936129924E-08	.1021655182E-07	.1042732696E-07	.1070773886E-07	
	.1091852013E-07	.1119892205E-07	.1140970943E-07	.1169010137E-07	.1190089487E-07	.1218127683E-07

Table A2-37
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad N_p = 15,\ N_m = 9,\ N_B = 5$

α = 100.0

β = 1000.0

```
QLTSUM     = 0.                        .1089111025E-10  .3675940699E-10  .5722861604E-10  .8601872292E-10
              .1095870347E-09  .1416485522E-09  .1686620700E-09  .2043377823E-09  .2351359945E-09
              .2747621194E-09  .3096779967E-09  .3535842822E-09  .3929443730E-09  .4414541923E-09
              .4855787174E-09  .5390091628E-09  .5882121083E-09  .6468740835E-09  .7014632928E-09
              .7656616042E-09  .8259388685E-09  .8959723150E-09  .9622334626E-09  .1038394925E-08
              .1110929909E-08  .1193506436E-08  .1272599422E-08  .1361872318E-08  .1447801770E-08
              .1544046679E-08  .1637085443E-08  .1740572434E-08  .1840987821E-08  .1951981469E-08
              .2060035339E-08  .2178794809E-08  .2294743650E-08  .2421522776E-08  .2545617785E-08
              .2680655147E-08  .2813152310E-08  .2956711319E-08  .3097831487E-08  .3250140455E-08
              .3400129421E-08  .3561421642E-08  .3720510218E-08  .3891014041E-08  .4059428131E-08  .4239367034E-08

P* SUM     = 0.                        .3703704376E-21  .2962962356E-20  .9999996219E-20  .2370369190E-19
              .4629626867E-19  .7999994523E-19  .1270369397E-18  .1896294543E-18  .2699997132E-18
              .3703699549E-18  .4929623513E-18  .6399991012E-18  .8137024800E-18  .1016294625E-17
              .1249997835E-17  .1517034219E-17  .1819626034E-17  .2159995489E-17  .2540364791E-17
              .2962956053E-17  .3429991663E-17  .3943693625E-17  .4506284241E-17  .5119985712E-17
              .5787020243E-17  .6509609875E-17  .7289977119E-17  .8130343842E-17  .9032932404E-17
              .9999955003E-17  .1103366384E-16  .1213625111E-16  .1330994875E-16  .1455697945E-16
              .1587956489E-16  .1727992752E-16  .1876028952E-16  .2032287309E-16  .2196990006E-16
              .2370359332E-16  .2552617433E-16  .2743986564E-16  .2944688942E-16  .3154946788E-16
              .3374982272E-16  .3605017705E-16  .3845275209E-16  .4095977050E-16  .4357345446E-16  .4629602616E-16

G+P* SUM   = 0.                        .1089111025E-10  .3675940700E-10  .5722861605E-10  .8601872295E-10
              .1095870347E-09  .1416485523E-09  .1686620701E-09  .2043377825E-09  .2351359948E-09
              .2747621198E-09  .3096779972E-09  .3535842828E-09  .3929443738E-09  .4414541933E-09
              .4855737186E-09  .5390091643E-09  .5882121101E-09  .6468740856E-09  .7014632954E-09
              .7656616072E-09  .8259388719E-09  .8959723190E-09  .9622334671E-09  .1038394930E-08
              .1110929915E-08  .1193506442E-08  .1272599429E-08  .1361872326E-08  .1447801779E-08
              .1544046689E-08  .1637085454E-08  .1740572446E-08  .1840987834E-08  .1951981484E-08
              .2060035355E-08  .2178794827E-08  .2294743669E-08  .2421522796E-08  .2545617806E-08
              .2680655171E-08  .2813152336E-08  .2956711347E-08  .3097831516E-08  .3250140487E-08
              .3400129455E-08  .3561421678E-08  .3720510256E-08  .3891014382E-08  .4059428174E-08  .4239367081E-08
```

Table A2-38

Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad$ $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 1000.0$

$\beta = 1.0$

QLISUM	= 0.	.1678513556E-10	.6147896295E-10	.1180067830E-09	.2011733288E-09	
	.3009931163E-09	.4317444530E-09	.5829624434E-09	.7684774461E-09	.9774168226E-09	
	.1223229633E-08	.1494697470E-08	.1804945717E-08	.2142463764E-08	.2520103104E-08	
	.2926107188E-08	.3373096606E-08	.3849108798E-08	.4366570534E-08	.4913347668E-08	
	.5501705103E-08	.6119366760E-08	.6778463174E-08	.7466601837E-08	.8195802544E-08	
	.8953578260E-08	.9751858345E-08	.1057807999E-07	.1144409948E-07	.1233729430E-07	
	.1326946253E-07	.1422793565E-07	.1522446616E-07	.1624635123E-07	.1730530853E-07	
	.1838861083E-07	.1950795011E-07	.2065058308E-07	.2182818371E-07	.2302799975E-07	
	.2426159361E-07	.2551650998E-07	.2680410512E-07	.2811172559E-07	.2945102601E-07	
	.3080925885E-07	.3219808080E-07	.3360475362E-07	.3504093907E-07	.3649391092E-07	.3797533845E-07

P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374952272E-16	.3605017795E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16

G+P* SUM	= 0.	.1678513556E-10	.6147896295E-10	.1180067830E-09	.2011733289E-09	
	.3009931163E-09	.4317444530E-09	.5829624435E-09	.7684774463E-09	.9774168229E-09	
	.1223229633E-08	.1494697471E-08	.1804945718E-08	.2142463765E-08	.2520103105E-08	
	.2926107189E-08	.3373096607E-08	.3849108800E-08	.4366570536E-08	.4913347671E-08	
	.5501705106E-08	.6119366763E-08	.6778463178E-08	.7466601837E-08	.8195802549E-08	
	.8953578266E-08	.9751858351E-08	.1057807999E-07	.1144409949E-07	.1233729431E-07	
	.1326946254E-07	.1422793566E-07	.1522446617E-07	.1624635124E-07	.1730530854E-07	
	.1838861085E-07	.1950795012E-07	.2065058310E-07	.2182818373E-07	.2302799977E-07	
	.2426159363E-07	.2551651000E-07	.2680410515E-07	.2811172562E-07	.2945102604E-07	
	.3080925888E-07	.3219808083E-07	.3360475366E-07	.3504093911E-07	.3649391096E-07	.3797533850E-07

Table A2-39
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 1000.0$

$\beta = 10.0$

QLTSUM	= 0.	.9320589919E-10	.3852846252E-09	.9077888479E-09	.1605939703E-08	
	.2468007015E-08	.3435127377E-08	.4505146732E-08	.5631216568E-08	.6822050071E-08	
	.8039700603E-08	.9299855572E-08	.1056994705E-07	.1186975457E-07	.1316982768E-07	
	.1449229756E-07	.1580950374E-07	.1714492107E-07	.1847191540E-07	.1981472730E-07	
	.2114731181E-07	.2249434466E-07	.2383012012E-07	.2517956013E-07	.2651715534E-07	
	.2786796591E-07	.2920659723E-07	.3055818593E-07	.3189740548E-07	.3324943373E-07	
	.3458898554E-07	.3594125983E-07	.3728099763E-07	.3863340737E-07	.3997324755E-07	
	.4132572954E-07	.4266562431E-07	.4401814243E-07	.4535806449E-07	.4671059809E-07	
	.4805053183E-07	.4940306912E-07	.5074300562E-07	.5209553986E-07	.5343547402E-07	
	.5478800135E-07	.5612793026E-07	.5748044848E-07	.5882037047E-07	.6017287832E-07	.6151279245E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999965003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.9320589919E-10	.3852846252E-09	.9077888479E-09	.1605939703E-08	
	.2468007015E-08	.3435127378E-08	.4505146732E-08	.5631216568E-08	.6822050072E-08	
	.8039700603E-08	.9299855573E-08	.1056994705E-07	.1186975457E-07	.1316982768E-07	
	.1449229756E-07	.1580950374E-07	.1714492107E-07	.1847191540E-07	.1981472730E-07	
	.2114731181E-07	.2249434466E-07	.2383012012E-07	.2517956013E-07	.2651715535E-07	
	.2786796592E-07	.2920659724E-07	.3055818594E-07	.3189740549E-07	.3324943373E-07	
	.3458898555E-07	.3594125984E-07	.3728099764E-07	.3863340739E-07	.3997324756E-07	
	.4132572956E-07	.4266562435E-07	.4401814245E-07	.4535806451E-07	.4671059812E-07	
	.4805053186E-07	.4940306915E-07	.5074300565E-07	.5209553989E-07	.5343547405E-07	
	.5478800138E-07	.5612793030E-07	.5748044852E-07	.5882037051E-07	.6017287836E-07	.6151279249E-07

Table A2-40
Form 2, FTMP, intermittent, no restrictions

$T_{max} = 100$ min $\qquad N_p = 15,\ N_m = 9,\ N_B = 5$

$\alpha = 1000.0$

$\beta = 100.0$

QLTSUM	= 0.	.1693628182E-09	.5888433897E-09	.9732295138E-09	.1437181007E-08	
	.1830342875E-08	.2296009647E-08	.2689516549E-08	.3155238549E-08	.3548763607E-08	
	.4014476473E-08	.4408007012E-08	.4873708251E-08	.5267243777E-08	.5732933294E-08	
	.6126473787E-08	.6592151579E-08	.6985697037E-08	.7451363107E-08	.7844913526E-08	
	.8310567876E-08	.8704123255E-08	.9169765887E-08	.9563326225E-08	.1002895714E-07	
	.1042252243E-07	.1088814163E-07	.1128171188E-07	.1174731937E-07	.1214089457E-07	
	.1260649035E-07	.1300007050E-07	.1346565457E-07	.1385923967E-07	.1432481203E-07	
	.1471840208E-07	.1518396274E-07	.1557755773E-07	.1604310668E-07	.1643670662E-07	
	.1690224387E-07	.1729584875E-07	.1776137430E-07	.1815498412E-07	.1862049797E-07	
	.1901411272E-07	.1947961488E-07	.1987323457E-07	.2033872504E-07	.2073234966E-07	.2119782844E-07
P* SUM	= 0.	.3703704438E-21	.2962962455E-20	.9999996719E-20	.2370369348E-19	
	.4629627253E-19	.7999995323E-19	.1270369545E-18	.1896294796E-18	.2699997537E-18	
	.3703700166E-18	.4929624416E-18	.6399992292E-18	.8137026563E-18	.1016294862E-17	
	.1249998147E-17	.1517034624E-17	.1819626550E-17	.2159996137E-17	.2540365595E-17	
	.2962957041E-17	.3429992863E-17	.3943695071E-17	.4506285968E-17	.5119987760E-17	
	.5787022654E-17	.6509612696E-17	.7289980400E-17	.8130347636E-17	.9032936770E-17	
	.9999970003E-17	.1103366954E-16	.1213625758E-16	.1330995607E-16	.1455698770E-16	
	.1587957416E-16	.1727993789E-16	.1876030109E-16	.2032288596E-16	.2196991434E-16	
	.2370350913E-16	.2552619178E-16	.2743988485E-16	.2944691053E-16	.3154949102E-16	
	.3374984803E-16	.3605020469E-16	.3845278221E-16	.4095980327E-16	.4357349005E-16	.4629606474E-16
Q+P* SUM	= 0.	.1693628182E-09	.5888433898E-09	.9732295138E-09	.1437181007E-08	
	.1830342875E-08	.2296009647E-08	.2689516549E-08	.3155238549E-08	.3548763607E-08	
	.4014476473E-08	.4408007013E-08	.4873708251E-08	.5267243778E-08	.5732933295E-08	
	.6126473788E-08	.6592151581E-08	.6985697038E-08	.7451363109E-08	.7844913529E-08	
	.8310567879E-08	.8704123259E-08	.9169765891E-08	.9563326229E-08	.1002895714E-07	
	.1042252244E-07	.1088814164E-07	.1128171189E-07	.1174731938E-07	.1214089458E-07	
	.1260649036E-07	.1300007051E-07	.1346565458E-07	.1385923968E-07	.1432481205E-07	
	.1471840209E-07	.1518396275E-07	.1557755775E-07	.1604310670E-07	.1643670664E-07	
	.1690224389E-07	.1729584877E-07	.1776137433E-07	.1815498414E-07	.1862049800E-07	
	.1901411276E-07	.1947961492E-07	.1987323461E-07	.2033872508E-07	.2073234971E-07	.2119782849E-07

Table A2-41
Form 2, FTMP, intermittent, no restrictions

T_{max} = 100 min $\qquad$ $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 1000.0$

$\beta = 1000.0$

QLTSUM	= 0.	.1001051223E-09	.3357460331E-09	.5113447316E-09	.7512048059E-09	
	.9291871292E-09	.1171317011E-08	.1351529413E-08	.1595822327E-08	.1778160641E-08	
	.2024513969E-08	.2208877140E-08	.2457191440E-08	.2643481664E-08	.2893660804E-08	
	.3081783640E-08	.3333734897E-08	.3523599344E-08	.3777233455E-08	.3968751990E-08	
	.4223983180E-08	.4417071766E-08	.4673817739E-08	.4868395811E-08	.5126577729E-08	
	.5322558154E-08	.5582110586E-08	.5779439615E-08	.6040270481E-08	.6238867676E-08	
	.6500918169E-08	.6700716330E-08	.6963920838E-08	.7164855911E-08	.7429151923E-08	
	.7631162909E-08	.7896490921E-08	.8099519778E-08	.8365823191E-08	.8569814731E-08	
	.8837039747E-08	.9041941532E-08	.9310037052E-08	.9515799287E-08	.9784716804E-08	
	.9991292235E-08	.1026098573E-07	.1046832953E-07	.1073875536E-07	.1094682505E-07	.1121794185E-07
P* SUM	= 0.	.3703704438E-21	.2962962455E-20	.9999996719E-20	.2370369348E-19	
	.4629627253E-19	.7999995323E-19	.1270369545E-18	.1896294796E-18	.2699997537E-18	
	.3703700166E-18	.4929624416E-18	.6399992292E-18	.8137026563E-18	.1016294862E-17	
	.1249998147E-17	.1517034624E-17	.1819626550E-17	.2159996137E-17	.2540365595E-17	
	.2962957041E-17	.3429992863E-17	.3943695071E-17	.4506285968E-17	.5119987760E-17	
	.5787022654E-17	.6509612696E-17	.7289980400E-17	.8130347636E-17	.9032936770E-17	
	.9999970003E-17	.1103366954E-16	.1213625758E-16	.1330995607E-16	.1455698770E-16	
	.1587957416E-16	.1727993789E-16	.1876030109E-16	.2032298596E-16	.2196991434E-16	
	.2370360913E-16	.2552619178E-16	.2743988485E-16	.2944691053E-16	.3154949102E-16	
	.3374934803E-16	.3605020469E-16	.3845278221E-16	.4095980327E-16	.4357349005E-16	.4629606474E-16
Q+P* SUM	= 0.	.1001051223E-09	.3357460331E-09	.5113447316E-09	.7512048060E-09	
	.9291871293E-09	.1171317011E-08	.1351529413E-08	.1595822327E-08	.1778160641E-08	
	.2024513970E-08	.2208877141E-08	.2457191441E-08	.2643481665E-08	.2893660805E-08	
	.3081783641E-08	.3333734898E-08	.3523599345E-08	.3777233457E-08	.3968751993E-08	
	.4223983183E-08	.4417071769E-08	.4673817743E-08	.4868395815E-08	.5126577734E-08	
	.5322558160E-08	.5582110593E-08	.5779439623E-08	.6040270489E-08	.6238867685E-08	
	.6500918179E-08	.6700716341E-08	.6963920850E-08	.7164855924E-08	.7429151937E-08	
	.7631162925E-08	.7896490938E-08	.8099519797E-08	.8365823211E-08	.8569814753E-08	
	.8837039771E-08	.9041941557E-08	.9310037080E-08	.9515799317E-08	.9784716836E-08	
	.9991292269E-08	.1026098576E-07	.1046832957E-07	.1073875540E-07	.1094682509E-07	.1121794190E-07

Table A2-42
Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 10.0$

$\beta = 1.0$

QLTSUM	= 0.	.1025684778E-09	.3463617705E-09	.5375573469E-09	.7967854906E-09	
	.9984309987E-09	.1265446481E-08	.1472918817E-08	.1744262978E-08	.1954967498E-08	
	.2228700273E-08	.2441190052E-08	.2716234326E-08	.2929708638E-08	.3205469753E-08	
	.3419487296E-08	.3695637945E-08	.3909956051E-08	.4186316373E-08	.4400801760E-08	
	.4677273012E-08	.4891852525E-08	.5168380521E-08	.5383014023E-08	.5659569036E-08	
	.5874234508E-08	.6150800234E-08	.6365485599E-08	.6642053096E-08	.6856751727E-08	
	.7133316092E-08	.7348024356E-08	.7624582899E-08	.7839298802E-08	.8115850051E-08	
	.8330572497E-08	.8607115644E-08	.8821844034E-08	.9098378635E-08	.9313112639E-08	
	.9589638454E-08	.9804377888E-08	.1008089479E-07	.1029563955E-07	.1057214746E-07	
	.1078689750E-07	.1106339638E-07	.1127815166E-07	.1155464149E-07	.1176940200E-07	.1204588277E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946778E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1025684778E-09	.3463617705E-09	.5375573469E-09	.7967854906E-09	
	.9984309988E-09	.1265446481E-08	.1472918817E-08	.1744262978E-08	.1954967499E-08	
	.2228700273E-08	.2441190053E-08	.2716234327E-08	.2929708639E-08	.3205469754E-08	
	.3419487298E-08	.3695637947E-08	.3909956052E-08	.4186316376E-08	.4400801763E-08	
	.4677273015E-08	.4891852529E-08	.5168380525E-08	.5383014027E-08	.5659569041E-08	
	.5874234514E-08	.6150800241E-08	.6365485606E-08	.6642053104E-08	.6856751736E-08	
	.7133316102E-08	.7348024367E-08	.7624582912E-08	.7839298815E-08	.8115850065E-08	
	.8330572513E-08	.8607115661E-08	.8821844052E-08	.9098378656E-08	.9313112661E-08	
	.9589638477E-08	.9804377914E-08	.1008089481E-07	.1029563958E-07	.1057214749E-07	
	.1078689753E-07	.1106339641E-07	.1127815170E-07	.1155464153E-07	.1176940204E-07	.1204588282E-07

Table A2-43
Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $\qquad$ $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 10.0$

$\beta = 10.0$

QLTSUM	= 0.	.1103191906E-09	.3724885081E-09	.5759087126E-09	.8446818562E-09	
	.1048720143E-08	.1317538187E-08	.1521586873E-08	.1790395984E-08	.1994450336E-08	
	.2263250101E-08	.2467310081E-08	.2736100499E-08	.2940166104E-08	.3208947176E-08	
	.3413018406E-08	.3681790133E-08	.3885866986E-08	.4154629371E-08	.4358711844E-08	
	.4627464888E-08	.4831552980E-08	.5100296685E-08	.5304390394E-08	.5573124762E-08	
	.5777224087E-08	.6045949119E-08	.6250054059E-08	.6518769757E-08	.6722880308E-08	
	.6991586674E-08	.7195702836E-08	.7464399872E-08	.7668521642E-08	.7937209349E-08	
	.8141336727E-08	.8410015107E-08	.8614148090E-08	.8882817144E-08	.9086955731E-08	
	.9355615462E-08	.9559759651E-08	.9828410060E-08	.1003255985E-07	.1030120094E-07	
	.1050535633E-07	.1077398810E-07	.1097814908E-07	.1124677153E-07	.1145093811E-07	.1171955125E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703629549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1597956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017795E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1103191906E-09	.3724885081E-09	.5759087126E-09	.8446818562E-09	
	.1048720143E-08	.1317538187E-08	.1521586873E-08	.1790395984E-08	.1994450336E-08	
	.2263250102E-08	.2467310082E-08	.2736100499E-08	.2940166105E-08	.3208947177E-08	
	.3413018407E-08	.3681790135E-08	.3885866987E-08	.4154629373E-08	.4358711846E-08	
	.4627464891E-08	.4831552983E-08	.5100296689E-08	.5304390399E-08	.5573124767E-08	
	.5777224093E-08	.6045949126E-08	.6250054066E-08	.6518769765E-08	.6722880317E-08	
	.6991586684E-08	.7195702847E-08	.7464399884E-08	.7668521656E-08	.7937209364E-08	
	.8141336743E-08	.8410015124E-08	.8614148109E-08	.8882817164E-08	.9086955753E-08	
	.9355615496E-08	.9559759676E-08	.9828410087E-08	.1003255988E-07	.1030120097E-07	
	.1050535636E-07	.1077398813E-07	.1097814912E-07	.1124677157E-07	.1145093816E-07	.1171955130E-07

Table A2-44
Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $\quad$ $N_p = 15$, $N_m = 9$, $N_B = 5$

α = 10.0

β = 100.0

```
GLTSUM     = 0.                .1094173635E-09  .3674164879E-09  .5607208353E-09  .8218254998E-09
             .1015251232E-08  .1276350548E-08  .1469782477E-08  .1730872185E-08  .1924310150E-08
             .2185390247E-08  .2378834246E-08  .2639904732E-08  .2833354765E-08  .3094415643E-08
             .3287871706E-08  .3548922977E-08  .3742385070E-08  .4003426736E-08  .4196894857E-08
             .4457926919E-08  .4651401066E-08  .4912423527E-08  .5105903698E-08  .5366916560E-08
             .5560402753E-08  .5821406016E-08  .6014898230E-08  .6275891898E-08  .6469390130E-08
             .6730374203E-08  .6923878453E-08  .7184852934E-08  .7378363198E-08  .7639328089E-08
             .7832844366E-08  .8093799668E-08  .8287321957E-08  .8548267672E-08  .8741795971E-08
             .9002732100E-08  .9196266408E-08  .9457192954E-08  .9650733267E-08  .9911650231E-08
             .1010519655E-07  .1036610393E-07  .1055965625E-07  .1082055406E-07  .1101411238E-07  .1127500061E-07

P* SUM     = 0.                .3703704376E-21  .2962962356E-20  .9999996219E-20  .2370369190E-19
             .4629626867E-19  .7999994523E-19  .1270369397E-18  .1896294543E-18  .2699997132E-18
             .3703679549E-18  .4929623513E-18  .6399991012E-18  .8137024800E-18  .1016294625E-17
             .1249997835E-17  .1517034219E-17  .1819626034E-17  .2159995489E-17  .2540364791E-17
             .2962956053E-17  .3429991663E-17  .3943693625E-17  .4506284241E-17  .5119985712E-17
             .5787020243E-17  .6509609875E-17  .7289977119E-17  .8130343842E-17  .9032932404E-17
             .9999955003E-17  .1103366384E-16  .1213625111E-16  .1330994875E-16  .1455697945E-16
             .1587956439E-16  .1727992752E-16  .1876028952E-16  .2032287309E-16  .2196990006E-16
             .2370359332E-16  .2552617433E-16  .2743986564E-16  .2944688942E-16  .3154946788E-16
             .3374932272E-16  .3605017705E-16  .3845275209E-16  .4095977050E-16  .4357345446E-16  .4629602616E-16

Q+P* SUM   = 0.                .1094173635E-09  .3674164879E-09  .5607208353E-09  .8218254998E-09
             .1015251232E-08  .1276350549E-08  .1469782477E-08  .1730872185E-08  .1924310150E-08
             .2185390247E-08  .2378834247E-08  .2639904733E-08  .2833354766E-08  .3094415644E-08
             .3287871708E-08  .3548922979E-08  .3742385072E-08  .4003426738E-08  .4196894859E-08
             .4457926922E-08  .4651401069E-08  .4912423531E-08  .5105903702E-08  .5366916565E-08
             .5560402758E-08  .5821406023E-08  .6014898237E-08  .6275891906E-08  .6469390139E-08
             .6730374213E-08  .6923878464E-08  .7184852946E-08  .7378363211E-08  .7639328103E-08
             .7832844382E-08  .8093799685E-08  .8287321976E-08  .8548267692E-08  .8741795993E-08
             .9002732124E-08  .9196266433E-08  .9457192981E-08  .9650733296E-08  .9911650263E-08
             .1010519658E-07  .1036610397E-07  .1055965629E-07  .1082055410E-07  .1101411243E-07  .1127500066E-07
```

Table A2-45

Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $\qquad N_p = 15,\ N_m = 9,\ N_B = 5$

$\alpha =$ 10.0

$\beta =$ 1000.0

QLTSUM	= 0.	.5768901798E-10	.1936557468E-09	.2966699527E-09	.4410051131E-09	
	.5526725529E-09	.7059416980E-09	.8268220753E-09	.9895477708E-09	.1120128015E-08	
	.1292763034E-08	.1433463179E-08	.1616396191E-08	.1767574861E-08	.1961135720E-08	
	.2123095220E-08	.2327559867E-08	.2500550910E-08	.2716145929E-08	.2900372019E-08	
	.3127278858E-08	.3322900359E-08	.3561259243E-08	.3768397149E-08	.4018310717E-08	
	.4237050145E-08	.4498586806E-08	.4728980235E-08	.5002177289E-08	.5244247555E-08	
	.5529114075E-08	.5782857151E-08	.6079376652E-08	.6344764209E-08	.6652897119E-08	
	.6929878906E-08	.7249564844E-08	.7538070879E-08	.7869230771E-08	.8169173366E-08	
	.8511711387E-08	.8822987025E-08	.9176792399E-08	.9499283456E-08	.9864232113E-08	
	.1019730845E-07	.1057376455E-07	.1091828499E-07	.1130510235E-07	.1166041599E-07	.1205793935E-07
P* SUM	= 0.	.3703704438E-21	.2962962455E-20	.9999996719E-20	.2370369348E-19	
	.4629627253E-19	.7999995323E-19	.1270369545E-18	.1896294796E-18	.2699997537E-18	
	.3703700166E-18	.4929624416E-18	.6399992292E-18	.8137026563E-18	.1016294862E-17	
	.1249998147E-17	.1517034624E-17	.1819626550E-17	.2159996137E-17	.2540365595E-17	
	.2962957041E-17	.3429992863E-17	.3943695071E-17	.4506285968E-17	.5119987760E-17	
	.5787022654E-17	.6509612696E-17	.7289980400E-17	.8130347636E-17	.9032936770E-17	
	.9999970003E-17	.1103366954E-16	.1213625758E-16	.1330995607E-16	.1455698770E-16	
	.1587957416E-16	.1727993789E-16	.1876030109E-16	.2032288596E-16	.2196991434E-16	
	.2370360913E-16	.2552619178E-16	.2743988485E-16	.2944691053E-16	.3154949102E-16	
	.3374934803E-16	.3605020469E-16	.3845278221E-16	.4095980327E-16	.4357349005E-16	.4629606474E-16
Q+P* SUM	= 0.	.5768901798E-10	.1936557468E-09	.2966699527E-09	.4410051131E-09	
	.5526725530E-09	.7059416980E-09	.8268220754E-09	.9895477710E-09	.1120128015E-08	
	.1292763034E-08	.1433463179E-08	.1616396192E-08	.1767574862E-08	.1961135721E-08	
	.2123095221E-08	.2327559869E-08	.2500550911E-08	.2716145932E-08	.2900372021E-08	
	.3127278861E-08	.3322900367E-08	.3561259247E-08	.3768397154E-08	.4018310722E-08	
	.4237050151E-08	.4498586813E-08	.4728980242E-08	.5002177297E-08	.5244247564E-08	
	.5529114085E-08	.5782857162E-08	.6079376664E-08	.6344764223E-08	.6652897133E-08	
	.6929878922E-08	.7249564862E-08	.7538070897E-08	.7869230791E-08	.8169173388E-08	
	.8511711411E-08	.8822987051E-08	.9176792425E-08	.9499283486E-08	.9864232144E-08	
	.1019730848E-07	.1057376459E-07	.1091828503E-07	.1130510239E-07	.1166041604E-07	.1205793940E-07

Table A2-46
Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 100.0$

$\beta = 1.0$

QLTSUM	= 0.	.6888727954E-10	.2407702878E-09	.4107548244E-09	.6456433646E-09	
	.8736304215E-09	.1160950961E-08	.1435759188E-08	.1764479167E-08	.2075636416E-08	
	.2436070697E-08	.2774783252E-08	.3159055141E-08	.3518339867E-08	.3920307472E-08	
	.4294796051E-08	.4709790509E-08	.5095438629E-08	.5519968001E-08	.5913768638E-08	
	.6345249109E-08	.6744986073E-08	.7181519596E-08	.7585569310E-08	.8025768516E-08	
	.8432946462E-08	.8875800749E-08	.9285245220E-08	.9730020207E-08	.1014110559E-07	
	.1058726852E-07	.1099954126E-07	.1144570609E-07	.1185983769E-07	.1230772495E-07	
	.1272147766E-07	.1316988512E-07	.1358408696E-07	.1403286837E-07	.1444739496E-07	
	.1489644457E-07	.1531120600E-07	.1576044737E-07	.1617537865E-07	.1662475652E-07	
	.1703981067E-07	.1748928512E-07	.1790442820E-07	.1835397036E-07	.1876917784E-07	.1921876687E-07
F* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+F* SUM	= 0.	.6888727954E-10	.2407702879E-09	.4107548244E-09	.6456433646E-09	
	.8736304216E-09	.1160950961E-08	.1435759188E-08	.1764479167E-08	.2075636416E-08	
	.2436070698E-08	.2774783252E-08	.3159055142E-08	.3518339867E-08	.3920307473E-08	
	.4294796052E-08	.4709790510E-08	.5095438631E-08	.5519968003E-08	.5913768640E-08	
	.6345249112E-08	.6744986076E-08	.7181519600E-08	.7585569314E-08	.8025768521E-08	
	.8432946468E-08	.8875800756E-08	.9285245228E-08	.9730020215E-08	.1014110560E-07	
	.1058726853E-07	.1099954127E-07	.1144670610E-07	.1185983770E-07	.1230772496E-07	
	.1272147768E-07	.1316988514E-07	.1358408698E-07	.1403286839E-07	.1444739498E-07	
	.1489644460E-07	.1531120603E-07	.1576044740E-07	.1617537868E-07	.1662475655E-07	
	.1703981071E-07	.1748928516E-07	.1790442824E-07	.1835397040E-07	.1876917788E-07	.1921876691E-07

Table A2-47
Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 100.0$

$\beta = 10.0$

QLTSUM	= 0.	.1193756412E-09	.4210238244E-09	.7194889541E-09	.1074271203E-08	
	.1388220706E-08	.1747348635E-08	.2062519913E-08	.2421981414E-08	.2737250703E-08	
	.3096731181E-08	.3412010414E-08	.3771485220E-08	.4086767496E-08	.4446234699E-08	
	.4761519476E-08	.5120978925E-08	.5436266159E-08	.5795717845E-08	.6111007531E-08	
	.6470451453E-08	.6785743591E-08	.7145179750E-08	.7460474338E-08	.7819902736E-08	
	.8135199773E-08	.8494620411E-08	.8809919895E-08	.9169332774E-08	.9484634705E-08	
	.9844039826E-08	.1015934420E-07	.1051874157E-07	.1083404839E-07	.1119343800E-07	
	.1150874726E-07	.1186812912E-07	.1218344082E-07	.1254281492E-07	.1285812907E-07	
	.1321749542E-07	.1353281201E-07	.1389217060E-07	.1420748963E-07	.1456684048E-07	
	.1486216194E-07	.1524150504E-07	.1555682894E-07	.1591616429E-07	.1623149063E-07	.1659081823E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374932272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+F* SUM	= 0.	.1193756412E-09	.4210238244E-09	.7194889541E-09	.1074271203E-08	
	.1388220706E-08	.1747348635E-08	.2062519913E-08	.2421981414E-08	.2737250703E-08	
	.3096731182E-08	.3412010415E-08	.3771485221E-08	.4086767497E-08	.4446234700E-08	
	.4761519477E-08	.5120978927E-08	.5436266161E-08	.5795717847E-08	.6111007534E-08	
	.6470451456E-08	.6785743594E-08	.7145179754E-08	.7460474342E-08	.7819902741E-08	
	.8135199779E-08	.8494620417E-08	.8809919902E-08	.9169332782E-08	.9484634714E-08	
	.9844039836E-08	.1015934421E-07	.1051874158E-07	.1083404840E-07	.1119343801E-07	
	.1150874728E-07	.1186812913E-07	.1218344084E-07	.1254281494E-07	.1285812909E-07	
	.1321749544E-07	.1353281203E-07	.1389217063E-07	.1420748966E-07	.1456684051E-07	
	.1488216197E-07	.1524150508E-07	.1555682898E-07	.1591616433E-07	.1623149067E-07	.1659081828E-07

Table A2-48
Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 100.0$

$\beta = 100.0$

QLTSUM	= 0.	.1164907500E-09	.3923051121E-09	.6027456861E-09	.8833834306E-09	
	.1094070847E-08	.1374710512E-08	.1585404678E-08	.1866034350E-08	.2076734673E-08	
	.2357354324E-08	.2568060800E-08	.2848570432E-08	.3059383060E-08	.3339982676E-08	
	.3550701453E-08	.3831291054E-08	.4042015979E-08	.4322595567E-08	.4533326638E-08	
	.4813896215E-08	.5024633430E-08	.5305192998E-08	.5515936355E-08	.5796485915E-08	
	.6007235413E-08	.6287774967E-08	.6498530604E-08	.6779060154E-08	.6989821927E-08	
	.7270341476E-08	.7481109384E-08	.7761618933E-08	.7972392974E-08	.8252892525E-08	
	.8463672697E-08	.8744162252E-08	.8954948553E-08	.9235428113E-08	.9446220542E-08	
	.9726690110E-08	.9937488664E-08	.1021794824E-07	.1042875292E-07	.1070920251E-07	
	.1092001331E-07	.1120045291E-07	.1141126983E-07	.1169169944E-07	.1190252248E-07	.1218294212E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1164907500E-09	.3923051121E-09	.6027456861E-09	.8833834306E-09	
	.1094070847E-08	.1374710512E-08	.1585404678E-08	.1866034350E-08	.2076734673E-08	
	.2357354324E-08	.2568060800E-08	.2848570433E-08	.3059383061E-08	.3339982677E-08	
	.3550701454E-08	.3831291056E-08	.4042015981E-08	.4322595569E-08	.4533326641E-08	
	.4813896218E-08	.5024633434E-08	.5305193002E-08	.5515936359E-08	.5796485920E-08	
	.6007235419E-08	.6287774974E-08	.6498530611E-08	.6779060163E-08	.6989821937E-08	
	.7270341486E-08	.7481109395E-08	.7761618945E-08	.7972392988E-08	.8252892540E-08	
	.8463672713E-08	.8744162269E-08	.8954948572E-08	.9235428134E-08	.9446220564E-08	
	.9726690133E-08	.9937488690E-08	.1021794827E-07	.1042875295E-07	.1070920254E-07	
	.1092001334E-07	.1120045294E-07	.1141126987E-07	.1169169949E-07	.1190252253E-07	.1218294216E-07

Table A2-49

Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

α = 100.0

β = 1000.0

QLTSUM	= 0.	.1103037673E-10	.3750072268E-10	.5976987391E-10	.9191681896E-10	
	.1211234999E-09	.1614538555E-09	.2000812412E-09	.2510548698E-09	.3015396793E-09	
	.3655703258E-09	.4303034961E-09	.5097594567E-09	.5910868075E-09	.6882915902E-09	
	.7885144801E-09	.9057474563E-09	.1027123430E-08	.1166620477E-08	.1311363902E-08	
	.1475318039E-08	.1645600739E-08	.1836162748E-08	.2034114622E-08	.2253393671E-08	
	.2481103305E-08	.2731167553E-08	.2990682828E-08	.3273560027E-08	.3566888714E-08	
	.3884566802E-08	.4213677151E-08	.4568104836E-08	.4934926157E-08	.5328013494E-08	
	.5734436729E-08	.6168055690E-08	.6615933980E-08	.7091919014E-08	.7583068255E-08	
	.8103216844E-08	.8639416238E-08	.9205489440E-08	.9788482037E-08	.1040220503E-07	
	.1103369827E-07	.1169676087E-07	.1237842709E-07	.1309248432E-07	.1382596130E-07	.1459263383E-07
P* SUM	= 0.	.3703704438E-21	.2962962455E-20	.9999996719E-20	.2370369348E-19	
	.4629627253E-19	.7999995323E-19	.1270369545E-18	.1896294796E-18	.2699997537E-18	
	.3703700166E-18	.4929624416E-18	.6399992292E-18	.8137026563E-18	.1016294862E-17	
	.1249998147E-17	.1517034624E-17	.1819626550E-17	.2159996137E-17	.2540365595E-17	
	.2962957041E-17	.3429992863E-17	.3943695071E-17	.4506285968E-17	.5119987760E-17	
	.5787022654E-17	.6509612696E-17	.7289980400E-17	.8130347636E-17	.9032936770E-17	
	.9999970003E-17	.1103366954E-16	.1213625758E-16	.1330995607E-16	.1455698770E-16	
	.1587957416E-16	.1727993789E-16	.1876030109E-16	.2032288596E-16	.2196991434E-16	
	.2370360913E-16	.2552619178E-16	.2743982485E-16	.2944691053E-16	.3154949102E-16	
	.3374934803E-16	.3605020469E-16	.3845278221E-16	.4095980327E-16	.4357349005E-16	.4629606474E-16
Q+P* SUM	= 0.	.1103037673E-10	.3750072268E-10	.5976987392E-10	.9191681898E-10	
	.1211235000E-09	.1614538556E-09	.2000812413E-09	.2510548700E-09	.3015396796E-09	
	.3655703262E-09	.4303034966E-09	.5097594573E-09	.5910868083E-09	.6882915912E-09	
	.7885144813E-09	.9057474578E-09	.1027123431E-08	.1166620479E-08	.1311363905E-08	
	.1475318042E-08	.1645600742E-08	.1836162752E-08	.2034114626E-08	.2253393676E-08	
	.2481103310E-08	.2731167559E-08	.2990682835E-08	.3273560035E-08	.3566888723E-08	
	.3884566812E-08	.4213677162E-08	.4568104848E-08	.4934926170E-08	.5328013508E-08	
	.5734436745E-08	.6168055707E-08	.6615933999E-08	.7091919034E-08	.7583068277E-08	
	.8103216867E-08	.8639416263E-08	.9205489467E-08	.9788482067E-08	.1040220506E-07	
	.1103369830E-07	.1169676091E-07	.1237842713E-07	.1309248436E-07	.1382596134E-07	.1459263387E-07

Table A2-50

Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $\quad\quad$ N_p = 15, N_m = 9, N_B = 5

α = 1000.0

β = 1.0

GLTSUM =	0.	.1692091119E-10	.6219204102E-10	.1204031890E-09	.2066230149E-09	
	.3114451807E-09	.4493328361E-09	.6103266208E-09	.8083733995E-09	.1033042518E-08	
	.1297840619E-08	.1591936538E-08	.1928462556E-08	.2296248265E-08	.2708108580E-08	
	.3152586849E-08	.3642229901E-08	.4165340653E-08	.4734240423E-08	.5337031258E-08	
	.5985845713E-08	.6668619085E-08	.7397326846E-08	.8159766939E-08	.8967785019E-08	
	.9809066961E-08	.1069535204E-07	.1161423579E-07	.1257737120E-07	.1357228206E-07	
	.1461055232E-07	.1567964993E-07	.1679110448E-07	.1793234168E-07	.1911484929E-07	
	.2032602229E-07	.2157731727E-07	.2285610829E-07	.2417382974E-07	.2551784300E-07	
	.2689956786E-07	.2830635999E-07	.2974963092E-07	.3121673640E-07	.3271908499E-07	
	.3424403735E-07	.3580300349E-07	.3738335294E-07	.3899650071E-07	.4062982848E-07	.4229475921E-07
P* SUM =	0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956439E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374952272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM =	0.	.1692091119E-10	.6219204102E-10	.1204031890E-09	.2066230149E-09	
	.3114451808E-09	.4493328362E-09	.6103266209E-09	.8083733997E-09	.1033042518E-08	
	.1297840620E-08	.1591936538E-08	.1928462557E-08	.2296248266E-08	.2708108581E-08	
	.3152586851E-08	.3642229903E-08	.4165340655E-08	.4734240425E-08	.5337031260E-08	
	.5985845716E-08	.6668619089E-08	.7397326850E-08	.8159766943E-08	.8967785024E-08	
	.9809066967E-08	.1069535205E-07	.1161423580E-07	.1257737121E-07	.1357228207E-07	
	.1461055233E-07	.1567964994E-07	.1679110450E-07	.1793234169E-07	.1911484930E-07	
	.2032602230E-07	.2157731729E-07	.2285610831E-07	.2417382976E-07	.2551784302E-07	
	.2689956789E-07	.2830636002E-07	.2974963095E-07	.3121673642E-07	.3271908502E-07	
	.3424403739E-07	.3580300353E-07	.3738335298E-07	.3899650075E-07	.4062982853E-07	.4229475926E-07

Table A2-51

Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

α = 1000.0

β = 10.0

QLTSUM	= 0.	.9331248951E-10	.3857868172E-09	.9092191918E-09	.1608716436E-08	
	.2472615567E-08	.3441869882E-08	.4514375604E-08	.5643107860E-08	.6836846373E-08	
	.8057486724E-08	.9320800412E-08	.1059407705E-07	.1189719297E-07	.1320056555E-07	
	.1452643087E-07	.1584700269E-07	.1718586566E-07	.1851626370E-07	.1986255005E-07	
	.2119656008E-07	.2254908387E-07	.2388829731E-07	.2524123765E-07	.2658227795E-07	
	.2793659419E-07	.2927867461E-07	.3063377198E-07	.3197644290E-07	.3333198150E-07	
	.3467428595E-07	.3603077152E-07	.3737396263E-07	.3872988413E-07	.4007317798E-07	
	.4142917197E-07	.4277252058E-07	.4412855080E-07	.4547192676E-07	.4682797248E-07	
	.4817136012E-07	.4952740949E-07	.5087079986E-07	.5222684612E-07	.5357023409E-07	
	.5492627337E-07	.5626965603E-07	.5762568611E-07	.5896906179E-07	.6032508141E-07	.6166844914E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703629549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688947E-16	.3154946788E-16	
	.3374982272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.9331248951E-10	.3857868172E-09	.9092191918E-09	.1608716436E-08	
	.2472615568E-08	.3441869882E-08	.4514375604E-08	.5643107860E-08	.6836846373E-08	
	.8057486725E-08	.9320800413E-08	.1059407705E-07	.1189719297E-07	.1320056555E-07	
	.1452643087E-07	.1584700269E-07	.1718586566E-07	.1851626370E-07	.1986255006E-07	
	.2119656009E-07	.2254908387E-07	.2388829731E-07	.2524123766E-07	.2658227795E-07	
	.2793659419E-07	.2927867461E-07	.3063377199E-07	.3197644291E-07	.3333198151E-07	
	.3467428596E-07	.3603077153E-07	.3737396264E-07	.3872988415E-07	.4007317799E-07	
	.4142917199E-07	.4277252060E-07	.4412855082E-07	.4547192678E-07	.4682797250E-07	
	.4817136014E-07	.4952740952E-07	.5087079989E-07	.5222684615E-07	.5357023413E-07	
	.5492627340E-07	.5626965607E-07	.5762568615E-07	.5896906183E-07	.6032508146E-07	.6166844919E-07

Table A2-52

Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

α = 1000.0

β = 100.0

145

QLTSUM	= 0.	.1693988324E-09	.5889737532E-09	.9734634188E-09	.1437528953E-08	
	.1830798470E-08	.2296580098E-08	.2690194807E-08	.3156031691E-08	.3549664561E-08	
	.4015492309E-08	.4409130659E-08	.4874946776E-08	.5268590111E-08	.5734394502E-08	
	.6128042801E-08	.6593835465E-08	.6987488726E-08	.7453269664E-08	.7846927885E-08	
	.8312697100E-08	.8706360279E-08	.9172117771E-08	.9565785907E-08	.1003153168E-07	
	.1042520477E-07	.1089093882E-07	.1128461687E-07	.1175033920E-07	.1214402220E-07	
	.1260973282E-07	.1300342076E-07	.1346911967E-07	.1386281256E-07	.1432849976E-07	
	.1472219760E-07	.1518787309E-07	.1558157586E-07	.1604723965E-07	.1644094737E-07	
	.1690659945E-07	.1730031211E-07	.1776595248E-07	.1815967008E-07	.1862529875E-07	
	.1901902128E-07	.1948463826E-07	.1987836572E-07	.2034397101E-07	.2073770340E-07	.2120329699E-07
P* SUM	= 0.	.3703704376E-21	.2962962356E-20	.9999996219E-20	.2370369190E-19	
	.4629626867E-19	.7999994523E-19	.1270369397E-18	.1896294543E-18	.2699997132E-18	
	.3703699549E-18	.4929623513E-18	.6399991012E-18	.8137024800E-18	.1016294625E-17	
	.1249997835E-17	.1517034219E-17	.1819626034E-17	.2159995489E-17	.2540364791E-17	
	.2962956053E-17	.3429991663E-17	.3943693625E-17	.4506284241E-17	.5119985712E-17	
	.5787020243E-17	.6509609875E-17	.7289977119E-17	.8130343842E-17	.9032932404E-17	
	.9999955003E-17	.1103366384E-16	.1213625111E-16	.1330994875E-16	.1455697945E-16	
	.1587956489E-16	.1727992752E-16	.1876028952E-16	.2032287309E-16	.2196990006E-16	
	.2370359332E-16	.2552617433E-16	.2743986564E-16	.2944688942E-16	.3154946788E-16	
	.3374932272E-16	.3605017705E-16	.3845275209E-16	.4095977050E-16	.4357345446E-16	.4629602616E-16
Q+P* SUM	= 0.	.1693988324E-09	.5889737532E-09	.9734634188E-09	.1437528953E-08	
	.1830798470E-08	.2296580098E-08	.2690194807E-08	.3156031691E-08	.3549664562E-08	
	.4015492310E-08	.4409130659E-08	.4874946776E-08	.5268590111E-08	.5734394503E-08	
	.6128042802E-08	.6593835466E-08	.6987488728E-08	.7453269666E-08	.7846927888E-08	
	.8312697103E-08	.8706360282E-08	.9172117775E-08	.9565785911E-08	.1003153168E-07	
	.1042520477E-07	.1089093883E-07	.1128461687E-07	.1175033921E-07	.1214402221E-07	
	.1260973283E-07	.1300342077E-07	.1346911969E-07	.1386281257E-07	.1432849978E-07	
	.1472219761E-07	.1518787311E-07	.1558157588E-07	.1604723967E-07	.1644094739E-07	
	.1690659947E-07	.1730031213E-07	.1776595251E-07	.1815967011E-07	.1862529879E-07	
	.1901902132E-07	.1948463830E-07	.1987836576E-07	.2034397105E-07	.2073770344E-07	.2120329703E-07

Table A2-53

Form 2 restricted, FTMP, intermittent

T_{max} = 100 min $N_p = 15$, $N_m = 9$, $N_B = 5$

$\alpha = 1000.0$

$\beta = 1000.0$

QLTSUM =	0.	.2505663028E-09	.8351338507E-09	.1252725208E-08	.1837214894E-08	
	.2254804098E-08	.2839215941E-08	.3256802977E-08	.3841136998E-08	.4258721854E-08	
	.4842978071E-08	.5260560733E-08	.5844739158E-08	.6262319621E-08	.6846420293E-08	
	.7263998525E-08	.7848021454E-08	.8265597451E-08	.8849542657E-08	.9267116406E-08	
	.9850983908E-08	.1026855540E-07	.1085234521E-07	.1126991443E-07	.1185362658E-07	
	.1227119351E-07	.1285482802E-07	.1327239264E-07	.1385594953E-07	.1427351183E-07	
	.1485699111E-07	.1527455109E-07	.1585795279E-07	.1627551043E-07	.1685883456E-07	
	.1727538984E-07	.1785963643E-07	.1827718934E-07	.1886035840E-07	.1927790893E-07	
	.1986100049E-07	.2027854862E-07	.2086156269E-07	.2127910842E-07	.2186204503E-07	
	.2227958833E-07	.2286244749E-07	.2327998835E-07	.2386277009E-07	.2428030850E-07	.2486301283E-07
P* SUM =	0.	.7999955173E-22	.1279985658E-20	.6479891141E-20	.2047954131E-19	
	.4999360006E-19	.1036765164E-18	.1920724707E-18	.3276653206E-18	.5248535467E-18	
	.7999552015E-18	.1171207852E-17	.1658768528E-17	.2284713668E-17	.3073039066E-17	
	.4049559817E-17	.5242410261E-17	.6681043936E-17	.8397233517E-17	.1042457077E-16	
	.1279856648E-16	.1555665043E-16	.1873817132E-16	.2238439672E-16	.2653851299E-16	
	.3124562531E-16	.3655275754E-16	.4250885219E-16	.4916477038E-16	.5657329177E-16	
	.6478911453E-16	.7386885526E-16	.8387114900E-16	.9485614899E-16	.1068865269E-15	
	.1200264725E-15	.1343421939E-15	.1499018172E-15	.1667753865E-15	.1850348641E-15	
	.2047541300E-15	.2260089825E-15	.2488771373E-15	.2734382282E-15	.2997738068E-15	
	.3279573420E-15	.3581042206E-15	.3902717472E-15	.4245591434E-15	.4610575487E-15	.4998600199E-15
Q+P* SUM =	0.	.2505663028E-09	.8351338507E-09	.1252725208E-08	.1837214894E-08	
	.2254804098E-08	.2839215941E-08	.3256802978E-08	.3841136998E-08	.4258721854E-08	
	.4842978072E-08	.5260560734E-08	.5844739159E-08	.6262319623E-08	.6846420296E-08	
	.7263998529E-08	.7848021459E-08	.8265597458E-08	.8849542665E-08	.9267116417E-08	
	.9850983921E-08	.1026855541E-07	.1085234523E-07	.1126991445E-07	.1185362661E-07	
	.1227119354E-07	.1285482805E-07	.1327239268E-07	.1385594957E-07	.1427351189E-07	
	.1485699118E-07	.1527455117E-07	.1585795287E-07	.1627551052E-07	.1685883467E-07	
	.1727538996E-07	.1785963656E-07	.1827718949E-07	.1886035857E-07	.1927790912E-07	
	.1986100069E-07	.2027854885E-07	.2086156294E-07	.2127910869E-07	.2186204532E-07	
	.2227958866E-07	.2286244784E-07	.2327998874E-07	.2386277051E-07	.2428030897E-07	.2486301333E-07

Table A2-54

SIFT Case 1a

$T_{max} = 10$ hrs

$N_p = 10$, $N_B = 5$

QLTSUM	= 0.	.2003397758E-09	.6677378921E-09	.1001620719E-08	.1468960471E-08	
	.1802837633E-08	.2270119033E-08	.2603990523E-08	.3071213583E-08	.3405079394E-08	
	.3872244127E-08	.4206104252E-08	.4673210670E-08	.5007065100E-08	.5474113217E-08	
	.5807961946E-08	.6274951773E-08	.6608794797E-08	.7075726343E-08	.7409563645E-08	
	.7876436933E-08	.8210268511E-08	.8677083546E-08	.9010909393E-08	.9477665190E-08	
	.9811486297E-08	.1027818487E-07	.1061199923E-07	.1107863959E-07	.1141244819E-07	
	.1187903035E-07	.1221283319E-07	.1267935716E-07	.1301315424E-07	.1347962903E-07	
	.1381341133E-07	.1427981896E-07	.1461340447E-07	.1507995304E-07	.1541173367E-07	
	.1588002501E-07	.1621379894E-07	.1668003215E-07	.1701380027E-07	.1747997537E-07	
	.1781373767E-07	.1827985458E-07	.1861361115E-07	.1907967007E-07	.1941342073E-07	.1987942156E-07
P* SUM	= 0.	.3199985592E-16	.2559976967E-15	.8639883365E-15	.2047963140E-14	
	.3999910004E-14	.6911813376E-14	.1097565427E-13	.1638341020E-13	.2332755523E-13	
	.3199856004E-13	.4258989176E-13	.5529301412E-13	.7029988737E-13	.8780246831E-13	
	.1079927103E-12	.1310625632E-12	.1572039735E-12	.1866088841E-12	.2194692347E-12	
	.2559769611E-12	.2963239960E-12	.3407072688E-12	.3893037051E-12	.4423202268E-12	
	.4999437532E-12	.5673661993E-12	.6297794772E-12	.7023754951E-12	.7803461582E-12	
	.8638333678E-12	.9531790221E-12	.1048425016E-11	.1149813240E-11	.1257535582E-11	
	.1371783927E-11	.1492750155E-11	.1620626143E-11	.1755603766E-11	.1897874894E-11	
	.2047631393E-11	.2205065128E-11	.2370367958E-11	.2543731740E-11	.2725348327E-11	
	.2915409570E-11	.3114107312E-11	.3321633400E-11	.3538179670E-11	.3763937959E-11	.3999100100E-11
Q+P* SUM	= 0.	.2003398078E-09	.6677381481E-09	.1001621583E-08	.1468962519E-08	
	.1802341633E-08	.2270125944E-08	.2604001499E-08	.3071229966E-08	.3405102721E-08	
	.3872276126E-08	.4206146842E-08	.4673265963E-08	.5007135400E-08	.5474201020E-08	
	.5808069938E-08	.6275082836E-08	.6608951996E-08	.7075912952E-08	.7409783115E-08	
	.7876592910E-08	.8210564835E-08	.8677424249E-08	.9011298696E-08	.9478108510E-08	
	.9811986240E-08	.1027874723E-07	.1061262901E-07	.1107934196E-07	.1141322854E-07	
	.1187989423E-07	.1221378637E-07	.1268040559E-07	.1301430405E-07	.1348087756E-07	
	.1381478311E-07	.1428131171E-07	.1461522510E-07	.1508170955E-07	.1541563155E-07	
	.1588207264E-07	.1621600400E-07	.1668240252E-07	.1701634400E-07	.1748270072E-07	
	.1781555308E-07	.1828296878E-07	.1861693279E-07	.1908320825E-07	.1941718465E-07	.1988342066E-07

Table A2-55

SIFT Case 1a

T_{max} = 10 hrs

N_p = 9, N_B = 4

QLTSUM =	0.	.1555588820E-09	.5184884382E-09	.7777390512E-09	.1140426286E-08	
	.1399369406E-08	.1762714367E-08	.2021950009E-08	.2384752684E-08	.2643980845E-08	
	.3006741242E-08	.3265961918E-08	.3628680043E-08	.3887893232E-08	.4250569093E-08	
	.4509774790E-08	.4872408395E-08	.5131606595E-08	.5494197952E-08	.5753388653E-08	
	.6115937769E-08	.6375120966E-08	.6737627846E-08	.6996803539E-08	.7359268195E-08	
	.7618436375E-08	.7990858812E-08	.8240019478E-08	.8602399704E-08	.8861552852E-08	
	.9223390875E-08	.9493036501E-08	.9845332327E-08	.1010447043E-07	.1046672407E-07	
	.1072585464E-07	.1108806509E-07	.1134718913E-07	.1170935842E-07	.1196847392E-07	
	.1233060104E-07	.1248970900E-07	.1295179396E-07	.1321089437E-07	.1357293718E-07	
	.1383203005E-07	.1419403072E-07	.1445311603E-07	.1481507456E-07	.1507415233E-07	.1543606873E-07

P* SUM =	0.	.1199995998E-10	.4799968008E-10	.1079989200E-09	.1919974403E-09	
	.2999950002E-09	.4319913600E-09	.5879862803E-09	.7679795207E-09	.9719708403E-09	
	.1199960001E-08	.1451946751E-08	.1727930882E-08	.2027912122E-08	.2351890243E-08	
	.2699865004E-08	.3071836165E-08	.3447803485E-08	.3887766726E-08	.4331725649E-08	
	.4799580010E-08	.5291629571E-08	.5807574095E-08	.6347513338E-08	.6911447060E-08	
	.7499375024E-08	.8111296988E-08	.8747212713E-08	.9407121959E-08	.1009102448E-07	
	.1079892005E-07	.1153080841E-07	.1228668934E-07	.1306656259E-07	.1387042792E-07	
	.1469328509E-07	.1555013386E-07	.1642597399E-07	.1732580525E-07	.1824962738E-07	
	.1919744016E-07	.2016924333E-07	.2116503657E-07	.2218481992E-07	.2322859287E-07	
	.2429635525E-07	.2538810683E-07	.2650384737E-07	.2764357664E-07	.2880729439E-07	.2999500038E-07

Q+P* SUM =	0.	.1675588420E-09	.5664881183E-09	.8857379512E-09	.1332623726E-08	
	.1699864406E-08	.2194705727E-08	.2609956289E-08	.3152732265E-08	.3615951685E-08	
	.4206701242E-08	.4717908680E-08	.5356610925E-08	.5915805354E-08	.6502459336E-08	
	.7209539793E-08	.7944244559E-08	.8599410081E-08	.9381964678E-08	.1008511430E-07	
	.1091561778E-07	.1166675054E-07	.1254520194E-07	.1334431688E-07	.1427071525E-07	
	.1511781140E-07	.1609215580E-07	.1698723219E-07	.1800952166E-07	.1895257734E-07	
	.2002281092E-07	.2101384492E-07	.2213202167E-07	.2317103302E-07	.2433715199E-07	
	.2542413973E-07	.2663819996E-07	.2777316313E-07	.2903516366E-07	.3021810130E-07	
	.3152804119E-07	.3275895233E-07	.3411683062E-07	.3539571430E-07	.3680153005E-07	
	.3812338530E-07	.3958213754E-07	.4095696341E-07	.4245865120E-07	.4388144671E-07	.4543106910E-07

Table A2-56

SIFT Case 1a

T_{max} = 10 hrs

N_p = 8, N_B = 3

FOR NO. OF PROCESSORS = 10

QLTSUM	= 0.	[illegible]	[illegible]	.1249940664E-08	.1833134863E-08	
	.2249788353E-08	[illegible]	[illegible]	.3832591243E-08	[illegible]	
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
	.1482359941E-07	[illegible]	[illegible]	[illegible]	[illegible]	
	.1723742891E-07	.1781938507E-07	.1823593362E-07	.1881781256E-07	[illegible]	
	.1981615618E-07	.2023269140E-07	.2081441595E-07	.2123094450E-07	.2181259188E-07	
	.2222911374E-07	.2281068397E-07	.2322719913E-07	.2380869223E-07	.2422520068E-07	.2480661666E-07

P* SUM	= 0.	.1535797262E-30	.1965561025E-28	.3357902011E-27	.2515254001E-26	
	.1199208265E-25	.4296412873E-25	.1263793770E-24	.3217825677E-24	.7337917828E-24	
	.1533973835E-23	.2988886493E-23	.5495054737E-23	.9621647481E-23	.1616162030E-22	
	.2619208898E-22	.4114469784E-22	.6288674599E-22	.9381383378E-22	.1369547912E-21	
	.1960896486E-21	.2758814100E-21	.3820223878E-21	.5213958977E-21	.7022537904E-21	
	.9344114173E-21	.1229459387E-20	.1600994265E-20	.2064868136E-20	.2639458052E-20	
	.3345956097E-20	.4208680795E-20	.5255410639E-20	.6517740469E-20	.8031461465E-20	
	.9836965488E-20	.1197967454E-19	.1451049605E-19	.1748630479E-19	.2097045212E-19	
	.2503330330E-19	.2975280365E-19	.3521507432E-19	.4151503830E-19	.4875707758E-19	
	.5705572201E-19	.6653637077E-19	.7733604712E-19	.8960418710E-19	.1035034631E-18	.1192106426E-18

FOR NO. OF BUSES = 5

QLTSUM	= 0.	.5555438887E-12	.1851797777E-11	.2777660927E-11	.4073858522E-11	
	.4999671305E-11	.6295812609E-11	.7221575028E-11	.8517660043E-11	.9443372100E-11	
	.1073940083E-10	.1166506253E-10	.1296103497E-10	.1388664631E-10	.1518256247E-10	
	.1610812345E-10	.1740398334E-10	.1832949397E-10	.1962529757E-10	.2055075785E-10	
	.2184650518E-10	.2277191511E-10	.2406760617E-10	.2499296575E-10	.2628860054E-10	
	.2721390978E-10	.2850948830E-10	.2943474719E-10	.3073026944E-10	.3165547800E-10	
	.3295094399E-10	.3387610221E-10	.3517151193E-10	.3609661982E-10	.3739197329E-10	
	.3831703083E-10	.3961232805E-10	.4053733526E-10	.4183257622E-10	.4275753311E-10	
	.4405271781E-10	.4497762437E-10	.4627275283E-10	.4719760906E-10	.4849268127E-10	
	.4941748718E-10	.5071250314E-10	.5163725874E-10	.5293221846E-10	.5385692373E-10	.5515182721E-10

P* SUM	= 0.	.7999795176E-22	.1279934470E-20	.6479502359E-20	.2047790301E-19	
	.4999360045E-19	.1036640759E-18	.1920455824E-18	.3276128984E-18	.5247590815E-18	
	.7997952265E-18	.1170950215E-17	.1658370471E-17	.2284119720E-17	.3072178736E-17	
	.4048445101E-17	.5240732958E-17	.6678772767E-17	.8394211057E-17	.1042061019E-16	
	.1279344808E-16	.1555011801E-16	.1872992834E-16	.2237410226E-16	.2652577756E-16	
	.3123000641E-16	.3653375505E-16	.4248590361E-16	.4913724581E-16	.5654048877E-16	
	.6475025272E-16	.7382307077E-16	.8381738870E-16	.9479356458E-16	.1068138688E-15	
	.1199424834E-15	.1342455024E-15	.1497909309E-15	.1666486854E-15	.1848905931E-15	
	.2045903922E-15	.2258237311E-15	.2486681683E-15	.2732031724E-15	.2995101219E-15	
	.3276723042E-15	.3577749162E-15	.3899050641E-15	.4241517622E-15	.4606059337E-15	.4993604097E-15

Q+P* SUM	= 0.	.2505645222E-09	.8351325070E-09	.1252718325E-08	.1837208722E-08	
	.2254788024E-08	.2839200852E-08	.3256773628E-08	.3841108903E-08	.4258675144E-08	
	.4842932886E-08	.5260492577E-08	.5844672804E-08	.6262225935E-08	.6846328666E-08	
	.7263875227E-08	.7847900479E-08	.8265440459E-08	.8849388250E-08	.9266921637E-08	
	.9850791987E-08	.1026831877E-07	.1085211169E-07	.1126963187E-07	.1185334739E-07	
	.1227086093E-07	.1285449906E-07	.1327200597E-07	.1385556674E-07	.1427306700E-07	
	.1485655042E-07	.1527404402E-07	.1585745011E-07	.1627493703E-07	.1685826581E-07	
	.1727574606E-07	.1785899753E-07	.1827647111E-07	.1885964530E-07	.1927711216E-07	
	.1986020910E-07	.2027766925E-07	.2086068895E-07	.2127814238E-07	.2186108486E-07	
	.2227853155E-07	.2286139683E-07	.2327883678E-07	.2386162487E-07	.2427905806E-07	.2486176900E-07

Table A2-57

SIFT Case 1b; T_{max} = 10 hrs; N_p = 10, N_B = 5

FOR NO. OF PROCESSORS = 9

QLTSUM	= 0.	.2000052380E-09	.6666256407E-09	.9999490325E-09	.1466510962E-08	
	.1799825631E-08	.2266329099E-08	.2599635040E-08	.3066080057E-08	.3399377264E-08	
	.3865763843E-08	.4199052309E-08	.4665380461E-08	.4998660181E-08	.5464929919E-08	
	.5798200885E-08	.6264412220E-08	.6597674427E-08	.7063827371E-08	.7397080813E-08	
	.7863175377E-08	.8196420048E-08	.8662456245E-08	.8995692138E-08	.9461669979E-08	
	.9794897089E-08	.1026081659E-07	.1059403491E-07	.1105989607E-07	.1139310560E-07	
	.1185890844E-07	.1219210916E-07	.1265785370E-07	.1299104561E-07	.1345673185E-07	
	.1378991495E-07	.1425554290E-07	.1458871718E-07	.1505428686E-07	.1538745232E-07	
	.1585296373E-07	.1618612035E-07	.1665157352E-07	.1698472130E-07	.1745011623E-07	
	.1778325517E-07	.1824859187E-07	.1858172196E-07	.1904700045E-07	.1938012168E-07	.1984534196E-07

P* SUM	= 0.	.4607483936E-31	.5896918947E-29	.1007431047E-27	.7546365687E-27	
	.3597984576E-26	.1289078542E-25	.3791912141E-25	.9655021712E-25	.2201771632E-24	
	.4602841983E-24	.8968632360E-24	.1648912113E-23	.2887244830E-23	.4849843855E-23	
	.7859984335E-23	.1234735988E-22	.1887243934E-22	.2815428385E-22	.4110205317E-22	
	.5885043003E-22	.8279919135E-22	.1146571544E-21	.1564907265E-21	.2107772859E-21	
	.2804636219E-21	.3690296616E-21	.4805577106E-21	.6198074357E-21	.7922968146E-21	
	.1004389282E-20	.1263387296E-20	.1577632554E-20	.1956613079E-20	.2411077415E-20	
	.2953156132E-20	.3596490903E-20	.4356371337E-20	.5249879831E-20	.6296044637E-20	
	.7516001386E-20	.8933163286E-20	.1057340022E-19	.1246522698E-19	.1464000081E-19	
	.1713212858E-19	.1997928372E-19	.2322263316E-19	.2690707453E-19	.3108148385E-19	.3579897387E-19

FOR NO. OF BUSES = 4

QLTSUM	= 0.	.3333240277E-12	.1111070315E-11	.1666574807E-11	.2444277542E-11	
	.2999740261E-11	.3777399445E-11	.4332820393E-11	.5110436030E-11	.5665815209E-11	
	.6443387301E-11	.6998724714E-11	.7776253263E-11	.8331548912E-11	.9109033920E-11	
	.9664287808E-11	.1044172928E-10	.1099694141E-10	.1177433934E-10	.1232950971E-10	
	.1310686412E-10	.1366199273E-10	.1443930361E-10	.1499439047E-10	.1577165782E-10	
	.1632670293E-10	.1710392675E-10	.1765893012E-10	.1843611041E-10	.1899107204E-10	
	.1976820881E-10	.2032312870E-10	.2110022195E-10	.2165510010E-10	.2243214984E-10	
	.2298698624E-10	.2376399247E-10	.2431878714E-10	.2509574985E-10	.2565050279E-10	
	.2642742200E-10	.2698213321E-10	.2775900891E-10	.2831367839E-10	.2909051059E-10	
	.2964513835E-10	.3042192704E-10	.3097651307E-10	.3175325827E-10	.3230780258E-10	.3308450428E-10

P* SUM	= 0.	.3199221593E-16	.2559874570E-15	.8639364988E-15	.2047799310E-14	
	.3999510033E-14	.6910984008E-14	.1097411778E-13	.1638078906E-13	.2332285674E-13	
	.3199216097E-13	.4258052302E-13	.5527974539E-13	.7028161178E-13	.8777788706E-13	
	.1079603173E-12	.1310206299E-12	.1571505332E-12	.1865417170E-12	.2193858522E-12	
	.2558745908E-12	.2961995661E-12	.3405523928E-12	.3891246665E-12	.4421079640E-12	
	.4996938438E-12	.5620738449E-12	.6294394881E-12	.7019822750E-12	.7798936886E-12	
	.8633651933E-12	.9525882343E-12	.1047754239E-11	.1149054614E-11	.1256680749E-11	
	.1370824014E-11	.1491675761E-11	.1619427323E-11	.1754270014E-11	.1896395128E-11	
	.2045993943E-11	.2203257715E-11	.2368377685E-11	.2541545071E-11	.2722951076E-11	
	.2912786881E-11	.3111243651E-11	.3318512531E-11	.3534784647E-11	.3760251107E-11	.3995102999E-11

Q+P* SUM	= 0.	.2003385940E-09	.6677369670E-09	.1001616471E-08	.1468957287E-08	
	.1802829371E-08	.2270113409E-08	.2603978835E-08	.3071206874E-08	.3405066402E-08	
	.3872239222E-08	.4206093614E-08	.4673211994E-08	.5007062012E-08	.5474126731E-08	
	.5807973133E-08	.6274984970E-08	.6608828519E-08	.7075788252E-08	.7409629709E-08	
	.7876538116E-08	.8210378240E-08	.8677236101E-08	.9011075653E-08	.9477883745E-08	
	.9811723486E-08	.1027848259E-07	.1061232328E-07	.1107903416E-07	.1141287657E-07	
	.1187954001E-07	.1221338488E-07	.1268000168E-07	.1301384976E-07	.1348042068E-07	
	.1381427276E-07	.1428079857E-07	.1461465539E-07	.1508113688E-07	.1541499922E-07	
	.1588143715E-07	.1621530574E-07	.1668170091E-07	.1701557652E-07	.1748192969E-07	
	.1781581310E-07	.1828212504E-07	.1861601699E-07	.1908228849E-07	.1941618973E-07	.1988242157E-07

Table A2-58

SIFT Case 1b; T_{max} = 10 hrs; N_p = 9, N_B = 4

FOR NO. OF PROCESSORS = 8

QLTSUM	= 0.	.1555581172E-09	.5184874443E-09	.7777354151E-09	.1140622026E-08	
	.1399860451E-08	.1762704352E-08	.2021933228E-08	.2384734427E-08	.2643953751E-08	
	.3006712256E-08	.3265922026E-08	.3628637843E-08	.3887830055E-08	.4250511193E-08	
	.4509701844E-08	.4872332309E-08	.5131513396E-08	.5494101196E-08	.5753272717E-08	
	.6115817058E-08	.6374979811E-08	.6737482301E-08	.6996634681E-08	.7359094527E-08	
	.7618237333E-08	.7980654543E-08	.8239787771E-08	.8602162351E-08	.8861285998E-08	
	.9223617957E-08	.9482732021E-08	.9845021364E-08	.1010412584E-07	.1046637258E-07	
	.1072546747E-07	.1108767160E-07	.1134675690E-07	.1170891844E-07	.1196799414E-07	
	.1233011310E-07	.1258917920E-07	.1295125558E-07	.1321031209E-07	.1357234589E-07	
	.1383139279E-07	.1419338404E-07	.1445242133E-07	.1481437002E-07	.1507339770E-07	.1543530384E-07

P* SUM	= 0.	.1023905797E-31	.1310478851E-29	.2238869989E-28	.1677104316E-27	
	.7996320871E-27	.2864962757E-26	.8427651214E-26	.2145903698E-25	.4893706636E-25	
	.1023058365E-24	.1993467928E-24	.3665128665E-24	.6417768026E-24	.1078044889E-23	
	.1747187263E-23	.2744735925E-23	.4195301568E-23	.6258760271E-23	.9137261094E-23	
	.1308310554E-22	.1840754985E-22	.2549058103E-22	.3479171751E-22	.4686188518E-22	
	.6235641974E-22	.8204924598E-22	.1068482849E-21	.1378121389E-21	.1761681067E-21	
	.2233315770E-21	.2809268531E-21	.3508094581E-21	.4350899712E-21	.5361594456E-21	
	.6567164589E-21	.7997958464E-21	.9687991655E-21	.1167526946E-20	.1400212771E-20	
	.1671559243E-20	.1986775885E-20	.2351619024E-20	.2772433708E-20	.3256197707E-20	
	.3810567653E-20	.4443927360E-20	.5165438376E-20	.5985092827E-20	.6913768586E-20	.7963286833E-20

FOR NO. OF BUSES = 3

QLTSUM	= 0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	0.

P* SUM	= 0.	.1199971998E-10	.4799776013E-10	.1079924403E-09	.1919820811E-09	
	.2999650022E-09	.4319395241E-09	.5879039680E-09	.7678566538E-09	.9717959013E-09	
	.1199720033E-08	.1451627368E-08	.1727516228E-08	.2027384934E-08	.2351231806E-08	
	.2699055166E-08	.3070853334E-08	.3466624633E-08	.3886367382E-08	.4330079906E-08	
	.4797760522E-08	.5289407554E-08	.5805019324E-08	.6344594153E-08	.6908130361E-08	
	.7495626274E-08	.8107080210E-08	.8742490493E-08	.9401855444E-08	.1008517339E-07	
	.1079244264E-07	.1152366153E-07	.1227882838E-07	.1305794151E-07	.1386099924E-07	
	.1468799989E-07	.1553894179E-07	.1641382327E-07	.1731264264E-07	.1823539822E-07	
	.1918208834E-07	.2015271133E-07	.2114726550E-07	.2216574918E-07	.2320816070E-07	
	.2427449837E-07	.2536476051E-07	.2647894546E-07	.2761705154E-07	.2877907707E-07	.2996502037E-07

Q+P* SUM	= 0.	.1675578372E-09	.5664852044E-09	.8857278554E-09	.1332604107E-08	
	.1699825453E-08	.2194643876E-08	.2609837196E-08	.3152591081E-08	.3615749652E-08	
	.4206432289E-08	.4717549394E-08	.5356154071E-08	.5915222989E-08	.6601742999E-08	
	.7208757010E-08	.7943185643E-08	.8598138029E-08	.9380468578E-08	.1008335262E-07	
	.1091357838E-07	.1166438736E-07	.1254250163E-07	.1334122883E-07	.1426722489E-07	
	.1511386361E-07	.1608773475E-07	.1698227826E-07	.1800401780E-07	.1894645938E-07	
	.2001606060E-07	.2100639355E-07	.2212384974E-07	.2316206735E-07	.2432737182E-07	
	.2541346736E-07	.2662661339E-07	.2776058017E-07	.2902156108E-07	.3020339236E-07	
	.3151220144E-07	.3274189053E-07	.3409852108E-07	.3537606127E-07	.3678050659E-07	
	.3810589116E-07	.3955814455E-07	.4093136679E-07	.4243142156E-07	.4385247477E-07	.4540032421E-07

Table A2-59
SIFT Case 1b; T_{max} = 10 hrs; N_p = 8, N_B = 3

QLTSUM	= 0.	.2783460111E-09	.9277226072E-09	.1391613181E-08	.2040904831E-08	
	.2504795445E-08	.3154002172E-08	.3617892809E-08	.4267014635E-08	.4730905281E-08	
	.5379942228E-08	.5843832867E-08	.6492784957E-08	.6956675573E-08	.7605542829E-08	
	.8069433406E-08	.8718215850E-08	.9182106373E-08	.9830804027E-08	.1029469448E-07	
	.1094330737E-07	.1140719774E-07	.1205572588E-07	.1251961614E-07	.1316805956E-07	
	.1363194971E-07	.1428030843E-07	.1474419845E-07	.1539247248E-07	.1585636236E-07	
	.1650455174E-07	.1696844144E-07	.1761654619E-07	.1808043572E-07	.1872845585E-07	
	.1919234519E-07	.1984028073E-07	.2030416986E-07	.2095202083E-07	.2141590973E-07	
	.2206367616E-07	.2252756482E-07	.2317524672E-07	.2363913513E-07	.2428673253E-07	
	.2475062066E-07	.2539813358E-07	.2586202143E-07	.2650944989E-07	.2697333744E-07	.2762068146E-07
P* SUM	= 0.	.7999955173E-22	.1279985658E-20	.6479891141E-20	.2047954131E-19	
	.4999860006E-19	.1036765164E-18	.1920724707E-18	.3276653206E-18	.5248535467E-18	
	.7999552015E-18	.1171207852E-17	.1658768528E-17	.2284713668E-17	.3073039066E-17	
	.4049559817E-17	.5242410261E-17	.6681043936E-17	.8397233517E-17	.1042457077E-16	
	.1279856648E-16	.1555665043E-16	.1873817132E-16	.2238439672E-16	.2653851299E-16	
	.3124562531E-16	.3655275754E-16	.4250885219E-16	.4916477038E-16	.5657329177E-16	
	.6476911453E-16	.7386885526E-16	.8387104900E-16	.9485614899E-16	.1068865269E-15	
	.1200264725E-15	.1343421939E-15	.1499018172E-15	.1667753865E-15	.1850348641E-15	
	.2047541300E-15	.2260089825E-15	.2488771373E-15	.2734382282E-15	.2997738068E-15	
	.3279573420E-15	.3581042206E-15	.3902717472E-15	.4245591434E-15	.4610575487E-15	.4998600199E-15
Q+P* SUM	= 0.	.2783460111E-09	.9277276072E-09	.1391613181E-08	.2040904831E-08	
	.2504795445E-08	.3154002172E-08	.3617892809E-08	.4267014635E-08	.4730905281E-08	
	.5379942229E-08	.5843832868E-08	.6492784959E-08	.6956675575E-08	.7605542832E-08	
	.8069433410E-08	.8718215855E-08	.9182106380E-08	.9830804036E-08	.1029469449E-07	
	.1094330738E-07	.1140719775E-07	.1205572589E-07	.1251961617E-07	.1316805959E-07	
	.1363194974E-07	.1428030846E-07	.1474419849E-07	.1539247253E-07	.1585636241E-07	
	.1650455180E-07	.1696844152E-07	.1761654627E-07	.1808043581E-07	.1872845596E-07	
	.1919234531E-07	.1984028086E-07	.2030417001E-07	.2095202100E-07	.2141590992E-07	
	.2206367636E-07	.2252756505E-07	.2317524697E-07	.2363913540E-07	.2428673283E-07	
	.2475062099E-07	.2539813394E-07	.2586202182E-07	.2650945032E-07	.2697333790E-07	.2762068196E-07

Table A2-60

SIFT Case 1c

T_{max} = 10 hrs

N_p = 10, N_B = 5

QLTSUM =	0.	.2203409519E-09	.7344019149E-09	.1101619674E-08	.1615617563E-08	
	.1982830847E-08	.2496765676E-08	.2863974476E-08	.3377846258E-08	.3745050565E-08	
	.4258859315E-08	.4626059121E-08	.5139804853E-08	.5507000148E-08	.6020682875E-08	
	.6367373652E-08	.6901493388E-08	.7268679637E-08	.7782236398E-08	.8149418110E-08	
	.8662911908E-08	.9030089376E-08	.9543519924E-08	.9910692539E-08	.1042406045E-07	
	.1079122850E-07	.1130453350E-07	.1167169698E-07	.1218493906E-07	.1255209797E-07	
	.1306527716E-07	.1343243147E-07	.1394554779E-07	.1431269750E-07	.1482575095E-07	
	.1519289606E-07	.1570588666E-07	.1607302716E-07	.1658595492E-07	.1695309079E-07	
	.1746595573E-07	.1783308697E-07	.1834588909E-07	.1871301570E-07	.1922575503E-07	
	.1959287698E-07	.2010555353E-07	.2047267082E-07	.2098528460E-07	.2135239723E-07	.2186494826E-07
P* SUM =	0.	.3199985592E-16	.2559976967E-15	.8639883365E-15	.2047963140E-14	
	.3999910004E-14	.6911813376E-14	.1097565427E-13	.1638341020E-13	.2332705523E-13	
	.3199356004E-13	.4258989176E-13	.5529301412E-13	.7029988737E-13	.8780246831E-13	
	.1079927103E-12	.1310625632E-12	.1572039735E-12	.1866088841E-12	.2194692347E-12	
	.2559769611E-12	.2963239960E-12	.3407022688E-12	.3893037051E-12	.4423202268E-12	
	.4999437532E-12	.5623661993E-12	.6297794772E-12	.7023754951E-12	.7803461582E-12	
	.8638533678E-12	.9531790221E-12	.1048425016E-11	.1149813240E-11	.1257535582E-11	
	.1371783927E-11	.1492750155E-11	.1620676143E-11	.1755603766E-11	.1897874894E-11	
	.2047631393E-11	.2205065128E-11	.2370367958E-11	.2543731740E-11	.2725348327E-11	
	.2915409570E-11	.3114107312E-11	.3321633400E-11	.3538179670E-11	.3763937959E-11	.3999100100E-11
Q+P* SUM =	0.	.2203409839E-09	.7344021709E-09	.1101620538E-08	.1615619611E-08	
	.1982334847E-08	.2496772588E-08	.2863985451E-08	.3377862642E-08	.3745073892E-08	
	.4258391314E-08	.4626101711E-08	.5139866146E-08	.5507070448E-08	.6020770678E-08	
	.6367781644E-08	.6901624451E-08	.7268836841E-08	.7782423006E-08	.8149637580E-08	
	.8663167885E-08	.9030385400E-08	.9543860627E-08	.9911081842E-08	.1042450277E-07	
	.1079172845E-07	.1130509586E-07	.1167232576E-07	.1218564144E-07	.1255287831E-07	
	.1306614104E-07	.1343338465E-07	.1394659621E-07	.1431384732E-07	.1482700849E-07	
	.1519426785E-07	.1570737941E-07	.1607464778E-07	.1658771052E-07	.1695498866E-07	
	.1745800336E-07	.1783529203E-07	.1834825946E-07	.1871555943E-07	.1922848037E-07	
	.1959579239E-07	.2010866764E-07	.2047599246E-07	.2098882278E-07	.2135616117E-07	.2186894736E-07

Table A2-61

SIFT Case 1c

T_{max} = 10 hrs

N_p = 9, N_B = 4

QLTSUM	= 0.	.1688927775E-09	.5629308575E-09	.8444038982E-09	.1238396383E-08	
	.1519862044E-08	.1913808942E-08	.2195267219E-08	.2589168538E-08	.2870619427E-08	
	.3264475176E-08	.3545918673E-08	.3939728859E-08	.4221164959E-08	.4614929591E-08	
	.4896358290E-08	.5290077377E-08	.5571498671E-08	.5965172220E-08	.6246586104E-08	
	.6640214124E-08	.6921620595E-08	.7315203093E-08	.7596602146E-08	.7990139132E-08	
	.8271530763E-08	.8665022244E-08	.8946406448E-08	.9339852432E-08	.9621229206E-08	
	.1001462970E-07	.1029599904E-07	.1068935406E-07	.1097071596E-07	.1136402550E-07	
	.1164537996E-07	.1203864404E-07	.1231999105E-07	.1271320967E-07	.1299454923E-07	
	.1338772241E-07	.1366905451E-07	.1406218225E-07	.1434350689E-07	.1473658919E-07	
	.1501790637E-07	.1541094326E-07	.1569225296E-07	.1608524443E-07	.1636654667E-07	.1675949273E-07

P* SUM	= 0.	.1199995998E-10	.4799968008E-10	.1079989200E-09	.1919974403E-09	
	.2999950002E-09	.4319913600E-09	.5879862803E-09	.7679795207E-09	.9719708403E-09	
	.1199760001E-08	.1451946761E-08	.1727930882E-08	.2027912122E-08	.2351890243E-08	
	.2699865004E-08	.3071836165E-08	.3467803485E-08	.3887766726E-08	.4331725649E-08	
	.4799680010E-08	.5291629571E-08	.5807574095E-08	.6347513338E-08	.6911447060E-08	
	.7499375024E-08	.8111296988E-08	.8747212713E-08	.9407121958E-08	.1009102448E-07	
	.1079892005E-07	.1153080841E-07	.1228668934E-07	.1306656259E-07	.1387042792E-07	
	.1469828509E-07	.1555013386E-07	.1642597399E-07	.1732580525E-07	.1824962738E-07	
	.1919744016E-07	.2016924333E-07	.2116503667E-07	.2218481992E-07	.2322859287E-07	
	.2429635525E-07	.2538810683E-07	.2650384737E-07	.2764357664E-07	.2880729439E-07	.2999500038E-07

Q+P* SUM	= 0.	.1808927375E-09	.6109305376E-09	.9524028182E-09	.1430393823E-08	
	.1819857044E-08	.2345800302E-08	.2783253499E-08	.3357148059E-08	.3842590268E-08	
	.4464435177E-08	.4997865434E-08	.5667659741E-08	.6249077081E-08	.6966819834E-08	
	.7596223294E-08	.8361913542E-08	.9039302156E-08	.9852938946E-08	.1057831175E-07	
	.1143989413E-07	.1221325017E-07	.1312277719E-07	.1394411548E-07	.1490158619E-07	
	.1577090579E-07	.1677631923E-07	.1769361916E-07	.1874697439E-07	.1971225369E-07	
	.2081354975E-07	.2182680746E-07	.2297604340E-07	.2403727855E-07	.2523445342E-07	
	.2634366505E-07	.2758877790E-07	.2874596504E-07	.3003901492E-07	.3124417661E-07	
	.3258516256E-07	.3383829784E-07	.3522721891E-07	.3652832681E-07	.3796518206E-07	
	.3931426162E-07	.4079905008E-07	.4219610034E-07	.4372882107E-07	.4517384106E-07	.4675449311E-07

Table A2-62

SIFT Case 1c

T_{max} = 10 hrs

N_p = 8, N_B = 3

NO. [illegible]OCES[illegible] = 1[illegible]

QLTSUM =	0.	.2525097411E-11	.8416148953E-11	.1262442201E-10	.1851468083E-10	
	.2272288612E-10	.2861235241E-10	.3282048982E-10	.3870916377E-10	.4291723318E-10	
	.4880511497E-10	.5301311627E-10	.5890020610E-10	.6310813917E-10	.6899443722E-10	
	.7320230195E-10	.7908780840E-10	.8329560468E-10	.8918031974E-10	.9338804745E-10	
	.9927197128E-10	.1034796303E-09	.1093627631E-09	.1135703533E-09	.1194526953E-09	
	.1236602166E-09	.1295417679E-09	.1337492202E-09	.1396299811E-09	.1438373642E-09	
	.1497173348E-09	.1539246487E-09	.1598038292E-09	.1640110737E-09	.1698894643E-09	
	.1740966393E-09	.1799742402E-09	.1841813456E-09	.1900581569E-09	.1942651927E-09	
	.2001412146E-09	.2043481806E-09	.2102234134E-09	.2144303094E-09	.2203047532E-09	
	.2245115792E-09	.2303852342E-09	.2345919900E-09	.2404648564E-09	.2446715420E-09	.2505436200E-09

P* SUM =	0.	.1646592542E-30	.2107373066E-28	.3600191069E-27	.2696758267E-26	
	.1285752631E-25	.4606504039E-25	.1355015603E-24	.3450112349E-24	.7867671299E-24	
	.1644727501E-23	.3204705039E-23	.5891871990E-23	.1031652321E-22	.1732891840E-22	
	.2808402389E-22	.4411697364E-22	.6743006801E-22	.1005921285E-21	.1468510252E-21	
	.2102601937E-21	.2958199560E-21	.4096344612E-21	.5590850591E-21	.7530209190E-21	
	.1001967708E-20	.1318355135E-20	.1716764182E-20	.2214194815E-20	.2830355001E-20	
	.3587971829E-20	.4513125538E-20	.5635607281E-20	.6989301395E-20	.8612593012E-20	
	.1054880181E-19	.1284664267E-19	.1556071413E-19	.1875201531E-19	.2248849224E-19	
	.2684561429E-19	.3190698155E-19	.3776496395E-19	.4452137292E-19	.5228816638E-19	
	.6118818786E-19	.7135594051E-19	.8293839691E-19	.9609584526E-19	.1110027730E-18	.1278487885E-18

FOR NO. OF BUSES = 5

QLTSUM =	0.	.6110963682E-14	.2036970254E-13	.3055415436E-13	.4481225607E-13	
	.5499614246E-13	.6925360704E-13	.7943692801E-13	.9369375550E-13	.1038765111E-12	
	.1181327015E-12	.1283148917E-12	.1425704451E-12	.1527520700E-12	.1670069863E-12	
	.1771880459E-12	.1914423252E-12	.2016228195E-12	.2158764619E-12	.2260563909E-12	
	.2403093964E-12	.2504887602E-12	.2647411287E-12	.2749199272E-12	.2891716589E-12	
	.2993498923E-12	.3136009870E-12	.3237786552E-12	.3380291131E-12	.3482062162E-12	
	.3624560373E-12	.3726325753E-12	.3868817596E-12	.3970577325E-12	.4113062800E-12	
	.4214816879E-12	.4357295987E-12	.4459044414E-12	.4601517156E-12	.4703259933E-12	
	.4845726307E-12	.4947463435E-12	.5089923443E-12	.5191654921E-12	.5334108562E-12	
	.5435834390E-12	.5578281666E-12	.5680001845E-12	.5822442754E-12	.5924157285E-12	.6066591829E-12

P* SUM =	0.	.1171249129E-21	.1873949210E-20	.9486617773E-20	.2998160655E-19	
	.7319535235E-19	.1517738818E-18	.2811724412E-18	.4796551279E-18	.7682945170E-18	
	.1170971292E-17	.1714373877E-17	.2427998063E-17	.3344146641E-17	.4497929029E-17	
	.5927260898E-17	.7672863822E-17	.9778264873E-17	.1228979628E-16	.1525659506E-16	
	.1873060263E-16	.2276656443E-16	.2742202957E-16	.3275735051E-16	.3883568256E-16	
	.4572298363E-16	.5348801382E-16	.6220233506E-16	.7194031068E-16	.8277910515E-16	
	.9479868357E-16	.1080818115E-15	.1227140543E-15	.1387837772E-15	.1563821443E-15	
	.1756031189E-15	.1965434625E-15	.2193027350E-15	.2439832939E-15	.2706902941E-15	
	.2995316874E-15	.3306182224E-15	.3640634441E-15	.3999836931E-15	.4384981058E-15	
	.4797286135E-15	.5237999426E-15	.5708396135E-15	.6209779414E-15	.6743480342E-15	.7310857942E-15

Q+P* SUM =	0.	.2531208375E-11	.8436518657E-11	.1265497617E-10	.1855949312E-10	
	.2277788234E-10	.2868160617E-10	.3289992703E-10	.3880285801E-10	.4302111046E-10	
	.4892324884E-10	.5314143288E-10	.5904277897E-10	.6326089458E-10	.6916144870E-10	
	.7337949592E-10	.7927925841E-10	.8349723728E-10	.8939620849E-10	.9361411910E-10	
	.9951229941E-10	.1037301418E-09	.1096275317E-09	.1138453060E-09	.1197419058E-09	
	.1239596122E-09	.1298554224E-09	.1340730611E-09	.1399680822E-09	.1441856532E-09	
	.1500798856E-09	.1542973894E-09	.1601908337E-09	.1644082702E-09	.1703009270E-09	
	.1745182966E-09	.1804101663E-09	.1846274693E-09	.1905185526E-09	.1947357894E-09	
	.2006260869E-09	.2048432576E-09	.2107327698E-09	.2149498749E-09	.2208386027E-09	
	.2250556424E-09	.2309435863E-09	.2351605611E-09	.2410477218E-09	.2452646322E-09	.2511510104E-09

Table A2-63

SIFT Case 2; T_{max} = 10 hrs; N_p =10; N_B = 5

FOR NO. OF PROCESSORS = 9

QLTSUM	= 0.	.2020057530E-11	.6732928242E-11	.1009949911E-10	.1481177188E-10	
	.1817825232E-10	.2288992726E-10	.2625631721E-10	.3096739446E-10	.3433369384E-10	
	.3904417351E-10	.4241038228E-10	.4712026449E-10	.5048638257E-10	.5519566745E-10	
	.5856169478E-10	.6327038245E-10	.6663631897E-10	.7134440955E-10	.7471025520E-10	
	.7941774880E-10	.8278350352E-10	.8749040028E-10	.9083606400E-10	.9556236402E-10	
	.9892793669E-10	.1036336401E-09	.1069991216E-09	.1117042286E-09	.1150696189E-09	
	.1197741295E-09	.1231394286E-09	.1278433429E-09	.1312085507E-09	.1359118689E-09	
	.1392769854E-09	.1439797076E-09	.1473447325E-09	.1520468589E-09	.1554117924E-09	
	.1601133229E-09	.1634781648E-09	.1681790998E-09	.1715438501E-09	.1762441895E-09	
	.1796088481E-09	.1843085921E-09	.1876731590E-09	.1923723077E-09	.1957367828E-09	.2004353364E-09

P* SUM	= 0.	.4939864938E-31	.6322342687E-29	.1080114592E-27	.8090846801E-27	
	.3857598793E-26	.1382097775E-25	.4065549786E-25	.1035180068E-24	.2360676883E-24	
	.4935054691E-24	.9615984511E-24	.1767936534E-23	.3095668184E-23	.5199962082E-23	
	.8427441179E-23	.1323883547E-22	.2023509957E-22	.3018724100E-22	.4407010478E-22	
	.6310035990E-22	.8877893282E-22	.1229381330E-21	.1677937153E-21	.2260021242E-21	
	.3007231631E-21	.3956883348E-21	.5152750945E-21	.6645872624E-21	.8495418359E-21	
	.1076962447E-20	.1354679709E-20	.1691638688E-20	.2098013759E-20	.2585331063E-20	
	.3166598836E-20	.3856445828E-20	.4671268070E-20	.5629384221E-20	.6751199742E-20	
	.8059380132E-20	.9579033482E-20	.1133790256E-19	.1336656670E-19	.1569865372E-19	
	.1837106208E-19	.2142419358E-19	.2490219683E-19	.2885322163E-19	.3332968465E-19	.3838854653E-19

FOR NO. OF BUSES = 4

QLTSUM	= 0.	.3666549980E-14	.1222171936E-13	.1833222780E-13	.2688689659E-13	
	.3299692597E-13	.4155109421E-13	.4766064455E-13	.5621431227E-13	.6232338361E-13	
	.7087655083E-13	.7698514319E-13	.8553780994E-13	.9164592337E-13	.1001980897E-12	
	.1063057242E-12	.1148573901E-12	.1209645457E-12	.1295157112E-12	.1356223880E-12	
	.1441730532E-12	.1502792511E-12	.1588294159E-12	.1649351351E-12	.1734847996E-12	
	.1795900400E-12	.1881392043E-12	.1942439659E-12	.2027926300E-12	.2088969129E-12	
	.2174450767E-12	.2235488810E-12	.2320965446E-12	.2381998702E-12	.2467470337E-12	
	.2528498807E-12	.2613965441E-12	.2674989124E-12	.2760450757E-12	.2821469655E-12	
	.2906926287E-12	.2967940400E-12	.3053392032E-12	.3114401359E-12	.3199847991E-12	
	.3260852534E-12	.3346294166E-12	.3407293924E-12	.3492730557E-12	.3553725531E-12	.3639157164E-12

P* SUM	= 0.	.4259092891E-16	.3407188624E-15	.1149897237E-14	.2725613788E-14	
	.5323330556E-14	.9198483854E-14	.1460648430E-13	.2180271686E-13	.3104254075E-13	
	.4258128947E-13	.5667427090E-13	.7357676719E-13	.9354403480E-13	.1168313045E-12	
	.1436937813E-12	.1743866447E-12	.2091650484E-12	.2482841203E-12	.2919989630E-12	
	.3405646529E-12	.3942362412E-12	.4532687530E-12	.5179171882E-12	.5884365205E-12	
	.6650816983E-12	.7481076443E-12	.8377692555E-12	.9343214031E-12	.1038018933E-11	
	.1149116664E-11	.1267869392E-11	.1394531885E-11	.1529358886E-11	.1672605111E-11	
	.1824525254E-11	.1985373980E-11	.2155405929E-11	.2334875716E-11	.2524037931E-11	
	.2723147135E-11	.2932457868E-11	.3152224641E-11	.3382701941E-11	.3624144229E-11	
	.3876805940E-11	.4140941484E-11	.4416805243E-11	.4704651578E-11	.5004734819E-11	.5317309276E-11

Q+P* SUM	= 0.	.2023766671E-11	.6745490680E-11	.1011898124E-10	.1484138439E-10	
	.1821657258E-10	.2294067684E-10	.2631858434E-10	.3104541149E-10	.3442705976E-10	
	.3915763135E-10	.4254404169E-10	.4727937907E-10	.5067157253E-10	.5541269684E-10	
	.5881169429E-10	.6355962648E-10	.6696644857E-10	.7172220938E-10	.7513787655E-10	
	.7990248651E-10	.8332801901E-10	.8810249845E-10	.9153891632E-10	.9632428534E-10	
	.9977260843E-10	.1045698869E-09	.1080311348E-09	.1128413426E-09	.1163165347E-09	
	.1211406912E-09	.1246308469E-09	.1294699713E-09	.1329761095E-09	.1378312210E-09	
	.1413543605E-09	.1462264781E-09	.1497676373E-09	.1546577797E-09	.1582179773E-09	
	.1631271627E-09	.1667074168E-09	.1716366636E-09	.1752379922E-09	.1801883185E-09	
	.1838117393E-09	.1887841630E-09	.1924306936E-09	.1974262324E-09	.2010968902E-09	.2061165614E-09

Table A2-64

SIFT Case 2; T_{max} = 10 hrs; N_p = 9, N_B = 4

QLTSUM	= 0.	.1571139992E-11	.5236729020E-11	.7855136033E-11	.1152028819E-10	
	.1413859668E-10	.1780331203E-10	.2042152198E-10	.2408580060E-10	.2670391198E-10	
	.3036775394E-10	.3298576673E-10	.3664917211E-10	.3926708626E-10	.4293005513E-10	
	.4554787063E-10	.4921040307E-10	.5182811988E-10	.5549021597E-10	.5810783406E-10	
	.6176949306E-10	.6438701321E-10	.6804823681E-10	.7066565737E-10	.743264[illegible]184E-10	
	.7694376660E-10	.8060411802E-10	.8322134094E-10	.8688125638E-10	.8949838044E-10	
	.9315785997E-10	.9577488513E-10	.9943392883E-10	.1020508551E-09	.1057094630E-09	
	.1083262903E-09	.1119844626E-09	.1146011909E-09	.1182589275E-09	.1208755568E-09	
	.1245328579E-09	.1271493882E-09	.1308062539E-09	.1334226850E-09	.1370791153E-09	
	.1396954474E-09	.1433514424E-09	.1459676754E-09	.1496232351E-09	.1522393689E-09	.1558944935E-09
P* SUM	= 0.	.1097764395E-31	.1405007613E-29	.2400363742E-28	.1798074915E-27	
	.8573091142E-27	.3071607581E-26	.9035513228E-26	.2300678971E-25	.5246663946E-25	
	.1096844975E-24	.2137241579E-24	.3929462144E-24	.6880617591E-24	.1155792238E-23	
	.1873190325E-23	.2942676628E-23	.4497847016E-23	.6710105291E-23	.9796175905E-23	
	.1402655139E-22	.1973492907E-22	.2732869174E-22	.3730048664E-22	.5024095746E-22	
	.6685268365E-22	.8796538152E-22	.1145524214E-21	.1477487156E-21	.1888700304E-21	
	.2394337780E-21	.3011813408E-21	.3761019838E-21	.4664584079E-21	.5748139990E-21	
	.7040618271E-21	.8574554481E-21	.1038641564E-20	.1251694596E-20	.1501153216E-20	
	.1792058912E-20	.2129996607E-20	.2521137422E-20	.2972283610E-20	.3490915726E-20	
	.4085242087E-20	.4764250571E-20	.5537762811E-20	.6416490841E-20	.7412096236E-20	.8537251821E-20

FOR NO. OF BUSES = 3

QLTSUM	= 0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	
	0.	0.	0.	0.	0.	0.
P* SUM	= 0.	.1451965348E-10	.5807722782E-10	.1306706437E-09	.2322978222E-09	
	.3629566849E-09	.5226451522E-09	.7113611452E-09	.9291025866E-09	.1175867397E-08	
	.1451653498E-08	.1756458811E-08	.2090281258E-08	.2453118761E-08	.2844969242E-08	
	.3265830623E-08	.3715700827E-08	.4194577774E-08	.4702459389E-08	.5239343592E-08	
	.5805228308E-08	.6400111459E-08	.7023990966E-08	.7676864755E-08	.8358730745E-08	
	.9069586861E-08	.9809431027E-08	.1057826117E-07	.1137607520E-07	.1220287106E-07	
	.1305864665E-07	.1394339992E-07	.1485712877E-07	.1579983114E-07	.1677150495E-07	
	.1777214812E-07	.1880175857E-07	.1986033424E-07	.2094787304E-07	.2206437290E-07	
	.2320983174E-07	.2438424750E-07	.2558761808E-07	.2681994143E-07	.2808121545E-07	
	.2937143809E-07	.3069060726E-07	.3203872089E-07	.3341577690E-07	.3482177322E-07	.3625670778E-07
Q+P* SUM	= 0.	.1609079347E-10	.6331395685E-10	.1385257797E-09	.2438181104E-09	
	.3770952816E-09	.5404484643E-09	.7317826672E-09	.9531883872E-09	.1202571309E-08	
	.1482021252E-08	.1789444577E-08	.2126930430E-08	.2492385847E-08	.2887899298E-08	
	.3311378493E-08	.3764911230E-08	.4246405894E-08	.4757949605E-08	.5297451427E-08	
	.5866997802E-08	.6464498473E-08	.7092039202E-08	.7747530412E-08	.8433057190E-08	
	.9146530628E-08	.9890035146E-08	.1066148251E-07	.1146295646E-07	.1229236944E-07	
	.1315180451E-07	.1403917480E-07	.1495656270E-07	.1590188200E-07	.1687721441E-07	
	.1788047441E-07	.1891374303E-07	.1997493543E-07	.2106613197E-07	.2218524846E-07	
	.2333436460E-07	.2451139688E-07	.2571842434E-07	.2695336411E-07	.2821829457E-07	
	.2951113354E-07	.3083395870E-07	.3218468836E-07	.3356540013E-07	.3497401259E-07	.3641260227E-07

Table A2- 65

SIFT Case 2; T_{max} = 10 hrs; N_p = 8 , N_B = 3

APPENDIX 3

PROGRAM SOURCE LISTING

```
      PROGRAM C3GENF2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1       DEBUG=OUTPUT,TDPFL,TAPE7=TDPFL,FUNCFL,TAPE8=FUNCFL)
C
C  QLT ARRAY DIMENSION IS COMPARABLE TO QLT(I,J,K,51) IF A FOUR
C  DIMENSIONAL ARRAY WERE POSSIBLE, BUT QLT ARRAY IS ONLY
C  DIMENSIONED LARGE ENOUGH TO HOLD THE PROBABILITIES ASSOCIATED
C  WITH TWO CONTIGUOUS VECTOR SETS AT A TIME (NAMELY THE LARGEST TWO
C  CONTAINING 52 AND 60 UNIQUE VECTORS ASSOCIATED WITH NP,NM,NB FROM
C  15,9,5 DOWN TO 2,2,2).
C
C  THIS MAIN PROGRAM IN COMBINATION WITH LIBRARIES C3GF2SL AND C3GF2BL
C  CAN RUN MODELS:  FORM2 - NON-RESTRICTED
C                   FORM2A - RESTRICTED
C                   FORM2B - RESTRICTED
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,
     1              ALPHA1,BETA1,ALPHA2,BETA2
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),
     1              G2(3,51),AT(3,51),CT(3,51)
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),
     1               G2PWT(9,51),H2PWT(9,51)
      COMMON/DEBUGC/ DBFLCD,CDYDB(51),CBXYDB(51)
      REAL LAMP,LAMM,LAMB,LAMBG
      REAL INTGRAL
      REAL MINS,MSECS
      LOGICAL QNOEFCT,PNOEFCT,PSTCOM
      DIMENSION INTGRAL(3),QLTSUM(51),CPSTARL(51),CPSTSUM(51),PLT(51)
      DATA HRS,MINS,SECS,MSECS/1.0,60.0,3600.0,3.6E6/
      DATA QLTSUM/51*0.0/,INTGRAL/3*0.0/,CPSTSUM/51*0.0/
      DATA PRNTP,PRNTPST,PRNTQ/"  P(","  P*(","  Q("/
      DATA PQSUM,PPSTSUM,PTOTSUM/"  QLTSUM  ","  P* SUM  ","  Q+P* SUM"/
      DATA PRNTDY/" DY("/,PRNTBXY/"BXY("/
      DATA QNOEFCT/.FALSE./,PNOEFCT/.FALSE./,PSTCOM/.FALSE./
C  'N' CODE MEANS NO PRINTOUT
      DATA DBFLCD/1HN/,PRCODE/1HN/
                                                REV. 3-21-79
C
C  READ IN NO OF PROCESSORS, NO OF MEMORY UNITS, NO OF BUSES,
C  INTEGRATION STEP AND TIME T.
C  THEN READ IN NO OF SURVIVORS AND TIME BASE (HRS  ,MINS ,SECS ,MSECS)
C
      PRINT 1
    1 FORMAT(///" SPECIFY NP.LE.15,NM.LE.9,NB.LE.5 IN I2 FORMAT, STEP IN
     X F6.2 FORMAT AND TMAX IN F6.1 FORMAT."/
     X" EXAMPLE:15,09,05,100.00,1000.0")
C
      READ(5,2) NP,NM,NB,STEP,TMAX
    2 FORMAT(3(I2,1X),F6.2,1X,F6.1)
C
      ITSTPS=TMAX/STEP + 0.5
C  ADD 1 TO INCLUDE TIME 0
      ITSTPS=ITSTPS+1
      IF(ITSTPS.LE.51) GO TO 888
      PRINT *," ERROR - QLT ARRAY OVERFLOW. "
      STOP
```

```
C
  888 PRINT 3
    3 FORMAT(/" SPECIFY NO. OF SURVIVORS NPS,NMS,NBS IN I2 FORMAT, "
     X       "TIME BASE IN A5 FORMAT AND MODEL DESIRED(' ','A',OR 'B')."
     X/" EXAMPLE:11,05,03,HRS    A (OR MINS  OR SECS  OR MSECS)")
C
      READ(5,4) NPS,NMS,NBS,TBASE,MODEL
    4 FORMAT(3(I2,1X),A5,1X,A1)
C
      PRINT 5
    5 FORMAT(/" SPECIFY TRANSIENT PARAMETERS ALPHA1 AND BETA1 IN F6.1 FO
     1RMAT.")
      READ(5,6) ALPHA1,BETA1
    6 FORMAT(F6.1,1X,F6.1)
C
      PRINT 7
    7 FORMAT(/" TYPE P FOR PLT ARRAY PRINT; TYPE Q FOR CPSTARL AND QLT A
     1RRAYS PRINT - USE B FOR BOTH."/
     2" THEN TYPE F FOR DEBUG FILES CREATION, IF DESIRED.")
      READ(5,8) PRCODE,DBFLCD,STOPARM
    8 FORMAT(2A1,1X,E7.1)
      IF(STOPARM.EQ.0.0) STOPARM=1.0E-10
C
      PRINT 9,TMAX,TBASE,STEP,TBASE,NP,NM,NB,NPS,NMS,NBS
    9 FORMAT(5X," FOR ",F6.1,1X,A7," THE FOLLOWING PROBABILITIES "
     X  "WERE COMPUTED USING A STEP SIZE OF ",F6.2,1X,A5/
     X  ,5X," WHERE NP=",I2,",NM=",I2,",NB=",I2,
     X  "  AND  NPS=",I2,",NMS=",I2,",NBS=",I2,":"/)
C
      PRINT 10,MODEL,ALPHA1,BETA1
   10 FORMAT(5X," MODEL ",A1," USING ALPHA= ",F6.1," AND BETA= ",F6.1/)
C
C  CONVERT LAMDA VALUES AND DELTA VALUES TO PROPER
C  TIME BASE BY USING A TIME BASE CONVERSION FACTOR(TBCF)
C
      IF(ALPHA1.EQ.0.0) GO TO 12
C  CONVERT DELTA'S TO INTERMITTENT VALUES
      DELTAP=100.0
      DELTAM=100.0
      DELTAB=360.0
   12 CONTINUE
C
      TBCF=0.0
      IF(TBASE.EQ.5HHRS  ) TBCF=HRS
      IF(TBASE.EQ.5HMINS ) TBCF=MINS
      IF(TBASE.EQ.5HSECS ) TBCF=SECS
      IF(TBASE.EQ.5HMSECS) TBCF=MSECS
C
      IF(TBCF.EQ.0.0) PRINT *," ERROR: INCORRECT TIME BASE "
      LAMP=LAMP/TBCF
      LAMM=LAMM/TBCF
      LAMB=LAMB/TBCF
      LAMBG=LAMBG/TBCF
      DELTAP=DELTAP/TBCF
      DELTAM=DELTAM/TBCF
      DELTAB=DELTAB/TBCF
      DELTABG=DELTABG/TBCF
      ALPHA1=ALPHA1/TBCF
      BETA1=BETA1/TBCF
C  THIS MODEL, ALPHA1=ALPHA2,BETA1=BETA2
      ALPHA2=ALPHA1
      BETA2=BETA1
```

```
C
C   COMPUTE TIME DEPENDENT PORTIONS OF THE MODEL.
       REWIND 7
       REWIND 8
       CALL TDEPEND
C
C   COMPUTE INITIAL PROB FOR I=0,J=0,K=0: QLT(0,0,0)=0.0 FOR ALL T
C   THEREFORE PROB=CPSTARL FOR VECTOR(0,0,0) WHICH EQUALS P*(0,0,0)
C
       I=0
       J=0
       K=0
       T=0.0
       DO 15 IT=1,ITSTPS
       CPSTARL(IT)=CPSTAR(I,NP,1,IT)*CPSTAR(J,NM,2,IT)*CPSTAR(K,NB,3,IT)
       T=T+STEP
    15 CONTINUE
C
       IF(PRCODE.EQ.1HN) GO TO 20
       IF(PRCODE.EQ.1HQ) GO TO 18
       PRINT 11,PRNTP,I,J,K,(CPSTARL(IT),IT=1,ITSTPS)
    11 FORMAT(/5X,A4,I2,",",I1,",",I1,") = ",9(5(E16.10,1X)/19X),
      X  6(E16.10,1X))
C
    18 IF(PRCODE.EQ.1HP) GO TO 20
       PRINT 11,PRNTPST,I,J,K,(CPSTARL(IT),IT=1,ITSTPS)
       PRINT 11,PRNTQ,I,J,K,(QLT(1,IT),IT=1,ITSTPS)
       PRINT *," "
    20 CONTINUE
C
C   NSET ARRAY HOLDS THE NO OF UNIQUELY DEFINED L"S PER SET
C   FOR LATER USE BY THE I,J,K DIMENSION MAPPING SUBROUTINE MAPDIM.
C   THE FIRST SET IS COMPRISED OF 1 L (NAMELY I=0,J=0,K=0)
       NSET(1)=1
C
C   COMPUTE MAXIMUM NUMBER OF FAILED PROCESSORS, MEMORY MODULES
C   AND BUSES ALLOWED (INCLUDING 0 FAILS)
C
       NPF=NP-NPS+1
       NMF=NM-NMS+1
       NBF=NB-NBS+1
C
       NPP1=NP+1
       NBP1=NB+1
       NMP1=NM+1
C
C   SET UP LOOP TO COMPUTE QLT IN SETS OF ISET CUBED PERMUTATIONS
C
       MAX=MAX0(NPF,NMF,NBF)
       MAXLAST=MAX
C
C   INITIALIZE QSUMSF(Q SUM SO FAR) AND CPSUMSF(P* SUM SO FAR) TO 0.0
       QSUMSF=0.0
       CPSUMSF=0.0
       DO 400 ISET=2,MAX
       ISET1=ISET
       ISET2=ISET
       ISET3=ISET
       IF(ISET1.GT.NBP1) ISET1=NBP1
       IF(ISET2.GT.NMP1) ISET2=NMP1
       IF(ISET3.GT.NPP1) ISET3=NPP1
```

```
C
C  INITIALIZE QLT INDEX N TO THE NUMBER OF VECTORS IN THE PREVIOUSLY
C  DEFINED SET + 1
      NUMPREV=NSET(ISET-1)
      N=NUMPREV+1
      IF(ISET.EQ.2) GO TO 55
C
C  POP VECTORS OFF QLT ARRAY WHICH WERE DEFINED TWO SETS AGO
C  BY MOVING THE VECTORS DEFINED IN THE PREVIOUS SET UP IN THE ARRAY.
C  IN THIS MANNER THE ONLY PROBABILITY VALUES STORED IN QLT ARRAY AT
C  ANY ONE TIME ARE THE PROBABILITIES OF VECTORS BEING DEFINED FOR
C  THE CURRENT SET AND VECTORS OF THE PREVIOUS SET.
C
      NPOP=NSET(ISET-2)
      DO 50 M=1,NUMPREV
      MM=NPOP+M
      DO 50 IT=1,ITSTPS
      QLT(M,IT)=QLT(MM,IT)
   50 CONTINUE
C
   55 CONTINUE
C  INITIALIZE TOTAL NUMBER OF L"S NOT PREVIOUSLY DEFINED IN THE
C    SET ISET  - NSTOT - TO 0.
C    (L REPRESENTS THE UNIQUE VECTOR I,J,K)
      NSTOT=0
C
C  BEGIN MAIN THREE LOOPS WHICH DEFINE L (VECTOR I,J,K)
C
      DO 300 KK=1,ISET1
      DO 200 JJ=1,ISET2
      DO 100 II=1,ISET3
C  DO NOT RECOMPUTE ANY PREVIOUSLY COMPUTED QLT(N)
      IF(II.LT.ISET .AND. JJ.LT.ISET .AND. KK.LT.ISET) GO TO 100
      I=II-1
      J=JJ-1
      K=KK-1
C
C  COMPUTE PERFECT COVERAGE PROBABILITIES FOR VECTORS FOR WHICH
C  QLT WILL NOT BE COMPUTED
C
      IF(KK.LE.NBF .AND. JJ.LE.NMF .AND. II.LE.NPF) GO TO 60
      IF(PNOEFCT) GO TO 100
      CPSTARL(1)=0.0
      DO 58 IT=2,ITSTPS
      CPSTARL(IT)=CPSTAR(I,NP,1,IT)*CPSTAR(J,NM,2,IT)*CPSTAR(K,NB,3,IT)
      CPSTSUM(IT)=CPSTSUM(IT)+CPSTARL(IT)
   58 CONTINUE
      IF(PRCODE.NE.1HP .AND. PRCODE.NE.1HN)
     1   PRINT 11,PRNTFST,I,J,K,(CPSTARL(IT),IT=1,ITSTPS)
      GO TO 100
   60 CONTINUE
C
C  COMPUTE SLAML WHERE L REPRESENTS VECTOR I,J,K
C
      SLAML=((NP-I)*LAMP)+((NM-J)*LAMM)+((NB-K)*LAMB)
C
C  QLT(N,1)=0.0 BECAUSE THIS REPRESENTS QLT(N) FOR T=0.0
      QLT(N,1)=0.0
      INTGRAL(1)=0.0
C  PERFECT COVERAGE PROBABILITY AT T=0.0 IS 0.0
      CPSTARL(1)=0.0
```

```
C
C   BEGIN MAIN INTEGRATION LOOP
C
        T=STEP
        DO 95 IT=1,ITSTPS
C
C   COMPUTE THE SUM
C
        IF(MODEL.EQ.1H ) CALL SUMMAT(II,JJ,KK,ISET,IT)
        IF(MODEL.EQ.1HA) CALL SUMRMA(II,JJ,KK,ISET,IT)
        IF(MODEL.EQ.1HB) CALL SUMRMB(II,JJ,KK,ISET,IT)
        IF(IT.EQ.1) GO TO 95
C
C   COMPUTE THE PERFECT COVERAGE PROBABILITIES FOR THE CURRENT
C   VECTOR
C
        CPSTARL(IT)=CPSTAR(I,NP,1,IT)*CPSTAR(J,NM,2,IT)*CPSTAR(K,NB,3,IT)
C
C   TRAPEZOIDAL RULE TO COMPUTE QLT(N,2)
C
        IF(IT.NE.2) GO TO 65
        CALL TRAPINT(SLAML,SUBINTG)
        INTGRAL(2)=SUBINTG
        PROB=EXP(-SLAML*T)*SUBINTG
        QLT(N,2)=PROB
        QLTSUM(2)=QLTSUM(2)+PROB
        T=T+STEP
        GO TO 95
C
C   PERFORM SIMPSON"S 1/3 INTEGRATION TECHNIQUE
C   TO COMPUTE QLT(N,IT),IT=3,ITSTPS
C
    65 CONTINUE
        CALL SIMPINT(IT,SLAML,SUBINTG)
        IF(IT.EQ.3) GO TO 80
        DO 75 IN=2,3
        INTGRAL(IN-1)=INTGRAL(IN)
    75 CONTINUE
    80 INTGRAL(3)=SUBINTG+INTGRAL(1)
C
C
C   COMPUTE QLT(N,IT)
C
        PROB=EXP(-SLAML*T)*INTGRAL(3)
        QLT(N,IT)=PROB
        QLTSUM(IT)=QLTSUM(IT)+PROB
        T=T+STEP
    95 CONTINUE
C
C   COMPUTE THE PROBABILITIES FOR THE CURRENT VECTOR BY SUBTRACTING
C   THE QLT FROM THE PERFECT COVERAGE PROBABILITIES
C
        IF(PRCODE.EQ.1HN) GO TO 99
        IF(PRCODE.EQ.1HQ) GO TO 98
        DO 97 IT=1,ITSTPS
        PLT(IT)=CPSTARL(IT)-QLT(N,IT)
    97 CONTINUE
C
        PRINT 11,PRNTP,I,J,K,(PLT(IT),IT=1,ITSTPS)
```

```
C
   98 IF(PRCODE.EQ.1HP) GO TO 99
      PRINT 11,PRNTPST,I,J,K,(CPSTARL(IT),IT=1,ITSTPS)
      PRINT 11,PRNTQ,I,J,K,(QLT(N,IT),IT=1,ITSTPS)
      PRINT *," "
   99 CONTINUE
C  WRITE CDY AND CBXY ARRAYS TO FUNCFL IF DBFLCD=C
      IF(DBFLCD.NE.1HC) GO TO 1000
      WRITE(8,11) PRNTDY,I,J,K,(CDYDB(IT),IT=1,ITSTPS)
      WRITE(8,11) PRNTBXY,I,J,K,(CBXYDB(IT),IT=1,ITSTPS)
 1000 CONTINUE
      N=N+1
      NSTOT=NSTOT+1
  100 CONTINUE
  200 CONTINUE
  300 CONTINUE
C  STORE TOTAL NUMBER OF UNIQUE L"S IN NSET ARRAY
      NSET(ISET)=NSTOT
C
C  ARE THE Q'S TOO SMALL TO AFFECT THE PROBABILITY?
C
      QSUMN=QLTSUM(ITSTPS)-QSUMSF
      IF(QSUMN.GE.STOPARM*QSUMSF) GO TO 310
      MAXLAST=ISET
      QNOEFCT=.TRUE.
  310 QSUMSF=QLTSUM(ITSTPS)
C
C  ARE THE P*'S TOO SMALL TO AFFECT THE PROBABILITY?
C
      IF(PNOEFCT) GO TO 390
C  HAVE ANY P*'S BEEN COMPUTED YET?
      CPSUMN=CPSTSUM(ITSTPS)-CPSUMSF
      IF(CPSUMN.NE.0.0) PSTCOM=.TRUE.
      IF(.NOT.PSTCOM) GO TO 390
      IF(CPSUMN.LT.STOPARM*AMAX1(QSUMSF,CPSUMSF))
     1  PNOEFCT=.TRUE.
      CPSUMSF=CPSTSUM(ITSTPS)
  390 IF(QNOEFCT) GO TO 410
  400 CONTINUE
  410 CONTINUE
C
C  COMPUTE PERFECT COVERAGE PROBABILITIES FOR ALL REMAINING
C  VECTORS FOR WHICH QLT WAS NOT COMPUTED
C  NOTE -  P*'S ARE NOT NECESSARILY MONOTONE DECREASING
C  FROM THE SETS COMPUTED ABOVE TO THE SETS TO BE COMPUTED.
```

```
C
      IF(PNOEFCT) GO TO 908
      QSUMSF=QLTSUM(ITSTPS)
      MAXN=MAX0(NPP1,NMP1,NBP1)
      MAXP1=MAXLAST+1
      DO 900 ISET=MAXP1,MAXN
      ISET1=ISET
      ISET2=ISET
      ISET3=ISET
      IF(ISET1.GT.NBP1) ISET1=NBP1
      IF(ISET2.GT.NMP1) ISET2=NMP1
      IF(ISET3.GT.NPP1) ISET3=NPP1
C  SAVE SUM SO FAR FOR LAST TIME STEP
      CPSUMSF=CPSTSUM(ITSTPS)
      DO 800 KK=1,ISET1
      DO 700 JJ=1,ISET2
      DO 600 II=1,ISET3
C  DO NOT COMPUTE CPSTARL FOR ANY PREVIOUSLY COMPUTED VECTOR
      IF(II.LT.ISET .AND. JJ.LT.ISET .AND. KK.LT.ISET) GO TO 600
      I=II-1
      J=JJ-1
      K=KK-1
      CPSTARL(1)=0.0
      DO 500 IT=2,ITSTPS
      CPSTARL(IT)=CPSTAR(I,NP,1,IT)*CPSTAR(J,NM,2,IT)*CPSTAR(K,NB,3,IT)
      CPSTSUM(IT)=CPSTSUM(IT)+CPSTARL(IT)
  500 CONTINUE
C
  600 CONTINUE
  700 CONTINUE
  800 CONTINUE
C  ARE THE P*'S TOO SMALL TO AFFECT THE PROBABILITY?
      CPSUMN=CPSTSUM(ITSTPS)-CPSUMSF
      IF(CPSUMN .LT. STOPARM*AMAX1(QSUMSF,CPSUMSF)) GO TO 905
C
  900 CONTINUE
  905 CONTINUE
      IF(PRCODE.EQ.1HF .OR. PRCODE.EQ.1HN) GO TO 908
      PRINT *,"       THE FINAL P* COMPUTED WAS:"
      PRINT 11,PRNTPST,I,J,K,(CPSTARL(IT),IT=1,ITSTPS)
C
C  PRINT THE SUM OF THE QLT'S, THE SUM OF THE P*'S,
C  AND THE SUM OF THE QLT'S+P*'S
C
  908 PRINT 910,PQSUM,(QLTSUM(IT),IT=1,ITSTPS)
  910 FORMAT(//5X,A10,"  = ",9(5(E16.10,1X)/19X),
     X  6(E16.10,1X))
C
      PRINT 910,PPSTSUM,(CPSTSUM(IT),IT=1,ITSTPS)
```

```
C
      DO 950 IT=1,ITSTPS
      QLTSUM(IT)=QLTSUM(IT)+CPSTSUM(IT)
  950 CONTINUE
      PRINT 910,PTOTSUM,(QLTSUM(IT),IT=1,ITSTPS)
      STOP
      END
      BLOCK DATA BLK1
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NEF,NSET(14),QLT(112,51)
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,
     1              ALPHA1,BETA1,ALPHA2,BETA2
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),
     1              G2(3,51),AT(3,51),CT(3,51)
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),
     1               G2PWT(9,51),H2PWT(9,51)
      REAL LAMP,LAMM,LAMB,LAMBG
      DATA LAMP,LAMM,LAMB,LAMBG/1.18E-4,1.18E-4,1.0E-6,0.18E-4/
      DATA DELTAP,DELTAM,DELTAB,DELTABG/2*3.6E3,3.6E4,3.6E2/
      DATA QLT/5712*0.0/,NSET/14*0/
      DATA EMLAM/153*0.0/,EMDEL/204*0.0/,EMLAM1/153*0.0/,EMLAM2/153*0.0/
      DATA G2/153*0.0/,AT/153*0.0/,CT/153*0.0/
      DATA G2WT/459*0.0/,H2WT/459*0.0/,G2PWT/459*0.0/,H2PWT/459*0.0/
      END
```

```
      SUBROUTINE TDEPEND                                                 TDEPEND
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,   TDEPEND
     1              ALPHA1,BETA1,ALPHA2,BETA2                            TDEPEND
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),   TDEPEND
     1              G2(3,51),AT(3,51),CT(3,51)                           TDEPEND
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                       TDEPEND
      COMMON/DEBUGC/ DBFLCD,CDYDB(51),CBXYDB(51)                         TDEPEND
      REAL LAMP,LAMM,LAMB,LAMBG                                          TDEPEND
C                                                                        TDEPEND
C   THIS SUBROUTINE COMPUTES PORTIONS OF THE MODEL THAT ARE DEPENDENT    TDEPEND
C   UPON TIME ONLY AND STORES THEM IN ARRAYS THAT CAN BE ACCESSED        TDEPEND
C   LATER.  IN THIS MANNER FUNCTIONS NEED NOT BE RECOMPUTED OVER AND     TDEPEND
C   OVER EVERY TIME THE STATE VECTOR CHANGES.                            TDEPEND
C                                                                        TDEPEND
C   COMPUTE EXP(-XLAM*T) AND EXP(-DELTA*T) FOR ALL T AND STORE IN ARRAYS TDEPEND
C   EMLAM AND EMDEL. (DO NOT CALL FUNCTION EXP IF -DELTA*T IS LESS THAN  TDEPEND
C   THE FUNCTION"S CAPACITY.  SINCE THE ARRAYS ARE INITIALIZED TO        TDEPEND
C   0.0, THE VALUE WILL DEFAULT TO 0.0.)                                 TDEPEND
C   THE 1ST INDEX INTO THE EMLAM AND EMDEL ARRAYS CORRESPOND TO:         TDEPEND
C       1: PROCESSOR RATES                                               TDEPEND
C       2: MEMORY UNIT RATES                                             TDEPEND
C       3: BUS RATES                                                     TDEPEND
C       4: BUS GUARDIAN UNIT RATES                                       TDEPEND
C                                                                        TDEPEND
      T=0.0                                                              TDEPEND
      DO 50 IT=1,ITSTPS                                                  TDEPEND
      EMLAM(1,IT)=EXP(-LAMP*T)                                           TDEPEND
      EMLAM(2,IT)=EXP(-LAMM*T)                                           TDEPEND
      EMLAM(3,IT)=EXP(-LAMB*T)                                           TDEPEND
      X=-DELTAP*T                                                        TDEPEND
      IF(X.GE.-675.84) EMDEL(1,IT)=EXP(X)                                TDEPEND
      X=-DELTAM*T                                                        TDEPEND
      IF(X.GE.-675.84) EMDEL(2,IT)=EXP(X)                                TDEPEND
      X=-DELTAB*T                                                        TDEPEND
      IF(X.GE.-675.84) EMDEL(3,IT)=EXP(X)                                TDEPEND
      X=-DELTABG*T                                                       TDEPEND
      IF(X.GE.-675.84) EMDEL(4,IT)=EXP(X)                                TDEPEND
C                                                                        TDEPEND
      X=-XLAM12(1,DELTAP)*T                                              TDEPEND
      IF(X.GE.-675.84) EMLAM1(1,IT)=EXP(X)                               TDEPEND
      X=-XLAM12(1,DELTAM)*T                                              TDEPEND
      IF(X.GE.-675.84) EMLAM1(2,IT)=EXP(X)                               TDEPEND
      X=-XLAM12(1,DELTAB)*T                                              TDEPEND
      IF(X.GE.-675.84) EMLAM1(3,IT)=EXP(X)                               TDEPEND
C                                                                        TDEPEND
      X=-XLAM12(2,DELTAP)*T                                              TDEPEND
      IF(X.GE.-675.84) EMLAM2(1,IT)=EXP(X)                               TDEPEND
      X=-XLAM12(2,DELTAM)*T                                              TDEPEND
      IF(X.GE.-675.84) EMLAM2(2,IT)=EXP(X)                               TDEPEND
      X=-XLAM12(2,DELTAB)*T                                              TDEPEND
      IF(X.GE.-675.84) EMLAM2(3,IT)=EXP(X)                               TDEPEND
C                                                                        TDEPEND
      T=T+STEP                                                           TDEPEND
   50 CONTINUE                                                           TDEPEND
```

```
C                                                                        TDEPEND
C   COMPUTE FUNCTION GTR2 FOR ALL COMBINATIONS OF DELTA1,DELTA2 FOR ALL TTDEPEND
C   AND STORE IN ARRAY G.                                                TDEPEND
C                                                                        TDEPEND
      IP=1                                                               TDEPEND
      IM=2                                                               TDEPEND
      IB=3                                                               TDEPEND
      DO 75 IT=1,ITSTPS                                                  TDEPEND
      G2(IP,IT)=GTR2(LAMP,DELTAP,IP,IT)                                  TDEPEND
      G2(IM,IT)=GTR2(LAMM,DELTAM,IM,IT)                                  TDEPEND
      G2(IB,IT)=GTR2(LAMB,DELTAB,IB,IT)                                  TDEPEND
   75 CONTINUE                                                           TDEPEND
C                                                                        TDEPEND
      IF(ALPHA1.GT.0.0) GO TO 90                                         TDEPEND
C   COMPUTE FUNCTION AFUNC AND STORE IN ARRAY AT FOR ALL T.              TDEPEND
C                                                                        TDEPEND
      DO 80 IT=1,ITSTPS                                                  TDEPEND
      AT(IP,IT)=AFUNC(LAMP,DELTAP,IP,IT)                                 TDEPEND
      AT(IM,IT)=AFUNC(LAMM,DELTAM,IM,IT)                                 TDEPEND
      AT(IB,IT)=AFUNC(LAMB,DELTAB,IB,IT)                                 TDEPEND
   80 CONTINUE                                                           TDEPEND
C   NOTE: FOR ALPHA1=0.0 AND BETA1=0.0,PERMANENT FAULT CASE,             TDEPEND
C   THERE IS NO NEED TO COMPUTE THE MARKOV MODEL.                        TDEPEND
      GO TO 450                                                          TDEPEND
C                                                                        TDEPEND
   90 CONTINUE                                                           TDEPEND
C                                                                        TDEPEND
C   COMPUTE FUNCTIONS ATR AND BTR WHICH HAVE BEEN INCORPORATED           TDEPEND
C   INTO ONE SUBROUTINE ABFUNCS AND STORE THEIR RATIO IN ARRAY CT        TDEPEND
C   FOR ALL T. (CT IS INFINITY AT TIME 0, THEREFORE IT MUST DEFAULT      TDEPEND
C   TO 0.0 AT TIME 0 AND THE FUNCTION PCOND WHICH USES CT WILL           TDEPEND
C   HANDLE THIS CASE.)                                                   TDEPEND
C   FUNCTION PX REQUIRES FUNCTION ATR ONLY.                              TDEPEND
C                                                                        TDEPEND
      DO 100 IT=2,ITSTPS                                                 TDEPEND
      CALL ABFUNCS(LAMP,DELTAP,IP,IT,ATR,BTR)                            TDEPEND
      AT(IP,IT)=ATR                                                      TDEPEND
      CT(IP,IT)=ATR/BTR                                                  TDEPEND
      CALL ABFUNCS(LAMM,DELTAM,IM,IT,ATR,BTR)                            TDEPEND
      AT(IM,IT)=ATR                                                      TDEPEND
      CT(IM,IT)=ATR/BTR                                                  TDEPEND
      CALL ABFUNCS(LAMB,DELTAB,IB,IT,ATR,BTR)                            TDEPEND
      AT(IB,IT)=ATR                                                      TDEPEND
      CT(IB,IT)=ATR/BTR                                                  TDEPEND
  100 CONTINUE                                                           TDEPEND
C                                                                        TDEPEND
C   CALL DEFF2B TO GENERATE THE MARKOV PROBABILITIES USED IN AORAP       TDEPEND
      CALL DEFF2B                                                        TDEPEND
```

```
                                                                       TDEPEND
C   WRITE CONTENTS OF CCMMON/INVAR/ TO TDPFL                           TDEPEND
  450 CONTINUE                                                         TDEPEND
      IF(DBFLCD.NE.1HF) GO TO 550                                      TDEPEND
      WRITE(7,499)                                                     TDEPEND
  499 FORMAT(1H )                                                      TDEPEND
      NAME=5HEMLAM                                                     TDEPEND
      WRITE(7,500) NAME,((EMLAM(IN,IT),IT=1,ITSTPS),IN=1,3)            TDEPEND
      NAME=5HEMDEL                                                     TDEPEND
      WRITE(7,510) NAME,((EMDEL(IN,IT),IT=1,ITSTPS),IN=1,4)            TDEPEND
      NAME=6HEMLAM1                                                    TDEPEND
      WRITE(7,500) NAME,((EMLAM1(IN,IT),IT=1,ITSTPS),IN=1,3)           TDEPEND
      NAME=6HEMLAM2                                                    TDEPEND
      WRITE(7,500) NAME,((EMLAM2(IN,IT),IT=1,ITSTPS),IN=1,3)           TDEPEND
      NAME=1HG                                                         TDEPEND
      WRITE(7,500) NAME,((G2(IN,IT),IT=1,ITSTPS),IN=1,3)               TDEPEND
      NAME=2HAT                                                        TDEPEND
      WRITE(7,500) NAME,((AT(IN,IT),IT=1,ITSTPS),IN=1,3)               TDEPEND
      NAME=2HCT                                                        TDEPEND
      WRITE(7,500) NAME,((CT(IN,IT),IT=1,ITSTPS),IN=1,3)               TDEPEND
                                                                       TDEPEND
  500 FORMAT(2X,A10,22(7(1X,E16.10)/12X))                              TDEPEND
  510 FORMAT(2X,A10,66(7(1X,E16.10)/12X))                              TDEPEND
C                                                                      TDEPEND
C                                                                      TDEPEND
C                                                                      TDEPEND
  550 RETURN                                                           TDEPEND
      END                                                              TDEPEND
      SUBROUTINE DEFF2A                                                DEFF2A
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG, DEFF2A
     1              ALPHA1,BETA1,ALPHA2,BETA2                          DEFF2A
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                     DEFF2A
      DIMENSION P1(9,51),P2(9,51),P3(9,51),P131(9),P132(9),P133(9)     DEFF2A
      DIMENSION PR123(9,51),PR13(9,51)                                 DEFF2A
C   COMPUTE THE NECESSARY MARKOV MODEL PROBABILITIES:                  DEFF2A
C   PR123  -  PROB. SYSTEM STARTING IN STATE 1 AT TIME TAU IS STILL    DEFF2A
C             IN STATE 1,2 OR 3 AT TIME T.                             DEFF2A
C   PR13   -  BETA2*PROB. SYSTEM STARTING IN STATE 1 AT TAU IS STILL   DEFF2A
C             (AGAIN) IN STATE 1 AT TIME T  +  BETA1*PROB. SYSTEM      DEFF2A
C             STARTING IN STATE 1 IS IN STATE 3 AT TIME T.             DEFF2A
C   **** FOR THESE PROBABILITIES DELTA1=DELTAY COMPONENT               DEFF2A
C        AND DELTA2=DELTAX COMPONENT. ****                             DEFF2A
C                                                                      DEFF2A
C   USING THE FOLLOWING STATE DIAGRAM:                                 DEFF2A
C                                                                      DEFF2A
C                            STATE 4                                   DEFF2A
C                           >       <                                  DEFF2A
C                          -         -                                 DEFF2A
C                         -           -                                DEFF2A
C               BETA2  -                 -  BETA1                      DEFF2A
C                  -                         -                         DEFF2A
C                 -                            -                       DEFF2A
C        STATE 1<-------BETA1---STATE 2---BETA2------>STATE 3          DEFF2A
C               -          >              <          -                 DEFF2A
C                ------ALPHA1            ALPHA2------                  DEFF2A
C                 -                                -                   DEFF2A
C                  -                              -                    DEFF2A
C               DELTA1                          DELTA2                 DEFF2A
C                    -                          -                      DEFF2A
C                     -                        -                       DEFF2A
C                       -                    -                         DEFF2A
C                         >                <                           DEFF2A
C                            STATE 5                                   DEFF2A
C                                                                      DEFF2A
```

```
C                                                                       DEFF2A
C  COMPUTE P1,P2,P3(1,ITU),ITU=1,ITSTPS                                 DEFF2A
C  NOTE - IF DELTAP=DELTAM, P'S(1,2,4,5 EQUAL)/ P'S(3,6 EQUAL) AND      DEFF2A
C                              P'S(7,8 EQUAL)                           DEFF2A
C                                                                       DEFF2A
      CALL MARKOV(DELTAP,DELTAP,1,P1,P2,P3,P131,P132,P133)              DEFF2A
C  COMPUTE P1,P2,P3(2,ITU),ITU=1,ITSTPS                                 DEFF2A
      IF(DELTAP.NE.DELTAM) GO TO 150                                    DEFF2A
      DO 125 ITU=1,ITSTPS                                               DEFF2A
      P1(2,ITU)=P1(1,ITU)                                               DEFF2A
      P2(2,ITU)=P2(1,ITU)                                               DEFF2A
      P3(2,ITU)=P3(1,ITU)                                               DEFF2A
  125 CONTINUE                                                          DEFF2A
      P131(2)=P131(1)                                                   DEFF2A
      P132(2)=P132(1)                                                   DEFF2A
      P133(2)=P133(1)                                                   DEFF2A
      GO TO 160                                                         DEFF2A
  150 CALL MARKOV(DELTAP,DELTAM,2,P1,P2,P3,P131,P132,P133)              DEFF2A
  160 CONTINUE                                                          DEFF2A
C  COMPUTE P1,P2,P3(3,ITU),ITU=1,ITSTPS                                 DEFF2A
      CALL MARKOV(DELTAP,DELTAB,3,P1,P2,P3,P131,P132,P133)              DEFF2A
C  COMPUTE P1,P2,P3(4,ITU),ITU=1,ITSTPS                                 DEFF2A
      IF(DELTAP.NE.DELTAM) GO TO 200                                    DEFF2A
      DO 175 ITU=1,ITSTPS                                               DEFF2A
      P1(4,ITU)=P1(1,ITU)                                               DEFF2A
      P2(4,ITU)=P2(1,ITU)                                               DEFF2A
      P3(4,ITU)=P3(1,ITU)                                               DEFF2A
  175 CONTINUE                                                          DEFF2A
      P131(4)=P131(1)                                                   DEFF2A
      P132(4)=P132(1)                                                   DEFF2A
      P133(4)=P133(1)                                                   DEFF2A
      GO TO 210                                                         DEFF2A
  200 CALL MARKOV(DELTAM,DELTAP,4,P1,P2,P3,P131,P132,P133)              DEFF2A
  210 CONTINUE                                                          DEFF2A
C  COMPUTE P1,P2,P3(5,ITU),ITU=1,ITSTPS                                 DEFF2A
      IF(DELTAP.NE.DELTAM) GO TO 250                                    DEFF2A
      DO 225 ITU=1,ITSTPS                                               DEFF2A
      P1(5,ITU)=P1(1,ITU)                                               DEFF2A
      P2(5,ITU)=P2(1,ITU)                                               DEFF2A
      P3(5,ITU)=P3(1,ITU)                                               DEFF2A
  225 CONTINUE                                                          DEFF2A
      P131(5)=P131(1)                                                   DEFF2A
      P132(5)=P132(1)                                                   DEFF2A
      P133(5)=P133(1)                                                   DEFF2A
      GO TO 260                                                         DEFF2A
  250 CALL MARKOV(DELTAM,DELTAM,5,P1,P2,P3,P131,P132,P133)              DEFF2A
  260 CONTINUE                                                          DEFF2A
C  COMPUTE P1,P2,P3(6,ITU),ITU=1,ITSTPS                                 DEFF2A
      IF(DELTAP.NE.DELTAM) GO TO 300                                    DEFF2A
      DO 275 ITU=1,ITSTPS                                               DEFF2A
      P1(6,ITU)=P1(3,ITU)                                               DEFF2A
      P2(6,ITU)=P2(3,ITU)                                               DEFF2A
      P3(6,ITU)=P3(3,ITU)                                               DEFF2A
  275 CONTINUE                                                          DEFF2A
      P131(6)=P131(3)                                                   DEFF2A
      P132(6)=P132(3)                                                   DEFF2A
      P133(6)=P133(3)                                                   DEFF2A
      GO TO 310                                                         DEFF2A
  300 CALL MARKOV(DELTAM,DELTAB,6,P1,P2,P3,P131,P132,P133)              DEFF2A
  310 CONTINUE                                                          DEFF2A
C  COMPUTE P1,P2,P3(7,ITU),ITU=1,ITSTPS                                 DEFF2A
      CALL MARKOV(DELTAB,DELTAP,7,P1,P2,P3,P131,P132,P133)              DEFF2A
```

```
C  COMPUTE P1,P2,P3(8,ITU),ITU=1,ITSTPS                               DEFF2A
      IF(DELTAP.NE.DELTAM) GO TO 350                                   DEFF2A
      DO 325 ITU=1,ITSTPS                                              DEFF2A
      P1(8,ITU)=P1(7,ITU)                                              DEFF2A
      P2(8,ITU)=P2(7,ITU)                                              DEFF2A
      P3(8,ITU)=P3(7,ITU)                                              DEFF2A
  325 CONTINUE                                                         DEFF2A
      P131(8)=P131(7)                                                  DEFF2A
      P132(8)=P132(7)                                                  DEFF2A
      P133(8)=P133(7)                                                  DEFF2A
      GO TO 360                                                        DEFF2A
  350 CALL MARKOV(DELTAB,DELTAM,8,P1,P2,P3,P131,P132,P133)             DEFF2A
C  COMPUTE P1,P2,P3(9,ITU),ITU=1,ITSTPS                               DEFF2A
  360 CALL MARKOV(DELTAB,DELTAB,9,P1,P2,P3,P131,P132,P133)             DEFF2A
C                                                                      DEFF2A
C                                                                      DEFF2A
C  DEFINE F2 AND F2P ARRAYS PR123 AND PR13                             DEFF2A
C  COMPUTE(PR123,PR13(IXY,ITU),IXY=1,9,ITU=1,ITSTPS                    DEFF2A
      DO 400 IXY=1,9                                                   DEFF2A
      DO 400 ITU=1,ITSTPS                                              DEFF2A
      P1ITU=P1(IXY,ITU)                                                DEFF2A
      P2ITU=P2(IXY,ITU)                                                DEFF2A
      P3ITU=P3(IXY,ITU)                                                DEFF2A
      PR123(IXY,ITU)=P1ITU+P2ITU+P3ITU                                 DEFF2A
      PR13(IXY,ITU)=P1ITU*P131(IXY) + P2ITU*P132(IXY) + P3ITU*P133(IXY) DEFF2A
  400 CONTINUE                                                         DEFF2A
C                                                                      DEFF2A
      WRITE(7,521)                                                     DEFF2A
  521 FORMAT(/18X,"T-TAU=0",10X,"T-TAU=STEP",5X,"T-TAU=2*STEP",5X,     DEFF2A
     1       ". . .")                                                  DEFF2A
      NAME=5HPR123                                                     DEFF2A
      DO 522 IXY=1,9                                                   DEFF2A
  522 WRITE(7,525) NAME,IXY,(PR123(IXY,IT),IT=1,ITSTPS)                DEFF2A
      WRITE(7,499)                                                     DEFF2A
  499 FORMAT(1H )                                                      DEFF2A
      NAME=4HPR13                                                      DEFF2A
      DO 524 IXY=1,9                                                   DEFF2A
  524 WRITE(7,525) NAME,IXY,(PR13(IXY,IT),IT=1,ITSTPS)                 DEFF2A
  525 FORMAT(2X,A5,"(",I1,")= ",8(7(1X,E16.10)/12X))                   DEFF2A
C                                                                      DEFF2A
      RETURN                                                           DEFF2A
      END                                                              DEFF2A
      SUBROUTINE DEFF2B                                                DEFF2B
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG, DEFF2B
     1              ALPHA1,BETA1,ALPHA2,BETA2                          DEFF2B
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),      DEFF2B
     1               G2PWT(9,51),H2PWT(9,51)                           DEFF2B
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                     DEFF2B
C                                                                      DEFF2B
C  COMPUTE CAPITAL A,B,C FOR F2 AND F2P ARRAYS                         DEFF2B
C  NOTE:  IF DELTAP=DELTAM, F2 AND F2P'S(1,2,4,5 EQUAL),(3,6 EQUAL)    DEFF2B
C         AND (7,8 EQUAL)                                              DEFF2B
C                                                                      DEFF2B
      CALL MRKOVB(DELTAP,DELTAP,1)                                     DEFF2B
      IF(DELTAP.NE.DELTAM) GO TO 150                                   DEFF2B
      DO 125 ITU=1,ITSTPS                                              DEFF2B
      G2WT(2,ITU)=G2WT(1,ITU)                                          DEFF2B
      H2WT(2,ITU)=H2WT(1,ITU)                                          DEFF2B
      G2PWT(2,ITU)=G2PWT(1,ITU)                                        DEFF2B
      H2PWT(2,ITU)=H2PWT(1,ITU)                                        DEFF2B
  125 CONTINUE                                                         DEFF2B
      GO TO 160                                                        DEFF2B
  150 CALL MRKOVB(DELTAP,DELTAM,2)                                     DEFF2B
```

```
  160 CONTINUE                                                     DEFF2B
      CALL MRKOVB(DELTAP,DELTAB,3)                                 DEFF2B
      IF(DELTAP.NE.DELTAM) GO TO 200                               DEFF2B
      DO 175 ITU=1,ITSTPS                                          DEFF2B
      G2WT(4,ITU)=G2WT(1,ITU)                                      DEFF2B
      H2WT(4,ITU)=H2WT(1,ITU)                                      DEFF2B
      G2PWT(4,ITU)=G2PWT(1,ITU)                                    DEFF2B
      H2PWT(4,ITU)=H2PWT(1,ITU)                                    DEFF2B
  175 CONTINUE                                                     DEFF2B
      GO TO 210                                                    DEFF2B
  200 CALL MRKOVB(DELTAM,DELTAP,4)                                 DEFF2B
  210 CONTINUE                                                     DEFF2B
      IF(DELTAP.NE.DELTAM) GO TO 250                               DEFF2B
      DO 225 ITU=1,ITSTPS                                          DEFF2B
      G2WT(5,ITU)=G2WT(1,ITU)                                      DEFF2B
      H2WT(5,ITU)=H2WT(1,ITU)                                      DEFF2B
      G2PWT(5,ITU)=G2PWT(1,ITU)                                    DEFF2B
      H2PWT(5,ITU)=H2PWT(1,ITU)                                    DEFF2B
  225 CONTINUE                                                     DEFF2B
      GO TO 260                                                    DEFF2B
  250 CALL MRKOVB(DELTAM,DELTAM,5)                                 DEFF2B
  260 CONTINUE                                                     DEFF2B
      IF(DELTAP.NE.DELTAM) GO TO 300                               DEFF2B
      DO 275 ITU=1,ITSTPS                                          DEFF2B
      G2WT(6,ITU)=G2WT(3,ITU)                                      DEFF2B
      H2WT(6,ITU)=H2WT(3,ITU)                                      DEFF2B
      G2PWT(6,ITU)=G2PWT(3,ITU)                                    DEFF2B
      H2PWT(6,ITU)=H2PWT(3,ITU)                                    DEFF2B
  275 CONTINUE                                                     DEFF2B
      GO TO 310                                                    DEFF2B
  300 CALL MRKOVB(DELTAM,DELTAB,6)                                 DEFF2B
  310 CONTINUE                                                     DEFF2B
      CALL MRKOVB(DELTAB,DELTAP,7)                                 DEFF2B
      IF(DELTAP.NE.DELTAM) GO TO 350                               DEFF2B
      DO 325 ITU=1,ITSTPS                                          DEFF2B
      G2WT(8,ITU)=G2WT(7,ITU)                                      DEFF2B
      H2WT(8,ITU)=H2WT(7,ITU)                                      DEFF2B
      G2PWT(8,ITU)=G2PWT(7,ITU)                                    DEFF2B
      H2PWT(8,ITU)=H2PWT(7,ITU)                                    DEFF2B
  325 CONTINUE                                                     DEFF2B
      GO TO 360                                                    DEFF2B
  350 CALL MRKOVB(DELTAB,DELTAM,8)                                 DEFF2B
  360 CALL MRKOVB(DELTAB,DELTAB,9)                                 DEFF2B
C                                                                  DEFF2B
C  WRITE G2WT,H2WT,G2PWT,H2PWT TO TDPFL                            DEFF2B
C                                                                  DEFF2B
      IBY=1                                                        DEFF2B
      IF(DELTAP.EQ.DELTAM) IBY=2                                   DEFF2B
      WRITE(7,540)                                                 DEFF2B
      NAME=4HG2WT                                                  DEFF2B
      DO 500 IXY=1,9,IBY                                           DEFF2B
      IF(IBY.EQ.2 .AND. IXY.EQ.5) GO TO 500                        DEFF2B
      WRITE(7,550) NAME,IXY,(G2WT(IXY,IT),IT=1,ITSTPS)             DEFF2B
  500 CONTINUE                                                     DEFF2B
C                                                                  DEFF2B
      WRITE(7,540)                                                 DEFF2B
      NAME=4HH2WT                                                  DEFF2B
      DO 510 IXY=1,9,IBY                                           DEFF2B
      IF(IBY.EQ.2 .AND. IXY.EQ.5) GO TO 510                        DEFF2B
      WRITE(7,550) NAME,IXY,(H2WT(IXY,IT),IT=1,ITSTPS)             DEFF2B
  510 CONTINUE                                                     DEFF2B
```

```
C                                                                       DEFF2B
      WRITE(7,540)                                                      DEFF2B
      NAME=5HG2PWT                                                      DEFF2B
      DO 520 IXY=1,9,IBY                                                DEFF2B
      IF(IBY.EQ.2 .AND. IXY.EQ.5) GO TO 520                             DEFF2B
      WRITE(7,550) NAME,IXY,(G2PWT(IXY,IT),IT=1,ITSTPS)                 DEFF2B
  520 CONTINUE                                                          DEFF2B
C                                                                       DEFF2B
      WRITE(7,540)                                                      DEFF2B
      NAME=5HH2PWT                                                      DEFF2B
      DO 530 IXY=1,9,IBY                                                DEFF2B
      IF(IBY.EQ.2 .AND. IXY.EQ.5) GO TO 530                             DEFF2B
      WRITE(7,550) NAME,IXY,(H2PWT(IXY,IT),IT=1,ITSTPS)                 DEFF2B
  530 CONTINUE                                                          DEFF2B
C                                                                       DEFF2B
  540 FORMAT(1H )                                                       DEFF2B
  550 FORMAT(2X,A5,"(",I1,")= ",8(7(1X,E16.10)/12X))                    DEFF2B
C                                                                       DEFF2B
      RETURN                                                            DEFF2B
      END                                                               DEFF2B
      SUBROUTINE MARKOV(DELTA1,DELTA2,IXY,P1,P2,P3,                     MARKOV
     1                  P131,P132,P133)                                 MARKOV
C                                                                       MARKOV
C  THIS SUBROUTINE CALLS EIGEN WHICH COMPUTES THE EIGENVALUES AND       MARKOV
C  EIGENVECTORS AND CONSTANTS NECESSARY TO COMPUTE STATE PROBABILITIES  MARKOV
C  1,2 AND 3.                                                           MARKOV
C                                                                       MARKOV
C  COMPUTE P1,P2,P3 FOR T-TAU = 0..ITSTPS                               MARKOV
      DIMENSION P1(9,51),P2(9,51),P3(9,51),P131(9),P132(9),P133(9)      MARKOV
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,  MARKOV
     1              ALPHA1,BETA1,ALPHA2,BETA2                           MARKOV
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      MARKOV
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),       MARKOV
     1               G2PWT(9,51),H2PWT(9,51)                            MARKOV
      COMMON/DEBUGC/ DBFLCD,CDYDB(51),CBXYDB(51)                        MARKOV
C                                                                       MARKOV
      CALL EIGEN(DELTA1,DELTA2)                                         MARKOV
C                                                                       MARKOV
C  SET INITIAL KNOWN CONDITIONS                                         MARKOV
      P1(IXY,1)=1.0                                                     MARKOV
      P2(IXY,1)=0.0                                                     MARKOV
      P3(IXY,1)=0.0                                                     MARKOV
C                                                                       MARKOV
      T=STEP                                                            MARKOV
      DO 150 ITAU=2,ITSTPS                                              MARKOV
      EIGL1T=0.0                                                        MARKOV
      X=EIGWR(1)*T                                                      MARKOV
      IF(X.GE.-675.84) EIGL1T=EXP(X)                                    MARKOV
      EIGL2T=0.0                                                        MARKOV
      X=EIGWR(2)*T                                                      MARKOV
      IF(X.GE.-675.84) EIGL2T=EXP(X)                                    MARKOV
      EIGL3T=0.0                                                        MARKOV
      X=EIGWR(3)*T                                                      MARKOV
      IF(X.GE.-675.84) EIGL3T=EXP(X)                                    MARKOV
C                                                                       MARKOV
      P1(IXY,ITAU)=EIGSD(1,1,1)*EIGL1T + EIGSD(1,2,1)*EIGL2T +          MARKOV
     1             EIGSD(1,3,1)*EIGL3T                                  MARKOV
      P2(IXY,ITAU)=EIGSD(2,1,1)*EIGL1T + EIGSD(2,2,1)*EIGL2T +          MARKOV
     1             EIGSD(2,3,1)*EIGL3T                                  MARKOV
      P3(IXY,ITAU)=EIGSD(3,1,1)*EIGL1T + EIGSD(3,2,1)*EIGL2T +          MARKOV
     1             EIGSD(3,3,1)*EIGL3T                                  MARKOV
C                                                                       MARKOV
      T=T+STEP                                                          MARKOV
  150 CONTINUE                                                          MARKOV
```

```
C                                                                         MARKOV
C   COMPUTE NON-TIME DEPENDENT TERMS OF PR13 PROBABILITY.                 MARKOV
C   COMPUTE COMMON TERMS (1-E**(EIGENVALUE*STEP))/EIGENVALUE              MARKOV
C   FOR EACH EIGENVALUE. THESE EIGENVALUES SHOULD BE NEGATIVE.            MARKOV
C                                                                         MARKOV
C   USING EIGENVALUE 1                                                    MARKOV
      EV1=EIGWR(1)                                                        MARKOV
      EIGL1ST=0.0                                                         MARKOV
      X=EV1*STEP                                                          MARKOV
      IF(X.GE.-675.84) EIGL1ST=EXP(X)                                     MARKOV
      EL1COMP=1.0-EIGL1ST                                                 MARKOV
      EV1INT=EL1COMP/ABS(EV1)                                             MARKOV
C   USING EIGENVALUE 2                                                    MARKOV
      EV2=EIGWR(2)                                                        MARKOV
      EIGL2ST=0.0                                                         MARKOV
      X=EV2*STEP                                                          MARKOV
      IF(X.GE.-675.84) EIGL2ST=EXP(X)                                     MARKOV
      EL2COMP=1.0-EIGL2ST                                                 MARKOV
      EV2INT=EL2COMP/ABS(EV2)                                             MARKOV
C   USING EIGENVALUE 3                                                    MARKOV
      EV3=EIGWR(3)                                                        MARKOV
      EIGL3ST=0.0                                                         MARKOV
      X=EV3*STEP                                                          MARKOV
      IF(X.GE.-675.84) EIGL3ST=EXP(X)                                     MARKOV
      EL3COMP=1.0-EIGL3ST                                                 MARKOV
      EV3INT=EL3COMP/ABS(EV3)                                             MARKOV
C                                                                         MARKOV
C   COMPUTE P(IST/KST), INTEGRATED OVER 1 STEP SIZE, WHERE KST IS THE     MARKOV
C   STARTING STATE AND IST IS THE CURRENT STATE FOR THE FOLLOWING         MARKOV
C   COMBINATIONS OF (IST/KST): (1/1),(3/1); (1/2),(3/2); (1/3),(3/3)      MARKOV
C   AND MULTIPLY BY APPROPRIATE BETA VALUE.                               MARKOV
C                                                                         MARKOV
      P131(IXY)=BETA2 * (EIGSD(1,1,1)*EV1INT + EIGSD(1,2,1)*EV2INT +      MARKOV
     1                   EIGSD(1,3,1)*EV3INT)                             MARKOV
     2          +BETA1 * (EIGSD(3,1,1)*EV1INT + EIGSD(3,2,1)*EV2INT +     MARKOV
     3                   EIGSD(3,3,1)*EV3INT)                             MARKOV
C                                                                         MARKOV
      P132(IXY)=BETA2 * (EIGSD(1,1,2)*EV1INT + EIGSD(1,2,2)*EV2INT +      MARKOV
     1                   EIGSD(1,3,2)*EV3INT)                             MARKOV
     2          +BETA1 * (EIGSD(3,1,2)*EV1INT + EIGSD(3,2,2)*EV2INT +     MARKOV
     3                   EIGSD(3,3,2)*EV3INT)                             MARKOV
C                                                                         MARKOV
      P133(IXY)=BETA2 * (EIGSD(1,1,3)*EV1INT + EIGSD(1,2,3)*EV2INT +      MARKOV
     1                   EIGSD(1,3,3)*EV3INT)                             MARKOV
     2          +BETA1 * (EIGSD(3,1,3)*EV1INT + EIGSD(3,2,3)*EV2INT +     MARKOV
     3                   EIGSD(3,3,3)*EV3INT)                             MARKOV
C                                                                         MARKOV
C   WRITE P ARRAYS TO TDPFL                                               MARKOV
      IF(DBFLCD.NE.1HF) GO TO 250                                         MARKOV
      WRITE(7,190)                                                        MARKOV
  190 FORMAT(1H )                                                         MARKOV
      DO 200 N1=1,3                                                       MARKOV
      WRITE(7,195)N1,IXY,EIGSD(N1,1,1),EIGWR(1),EIGSD(N1,2,1),EIGWR(2),   MARKOV
     1            EIGSD(N1,3,1),EIGWR(3)                                  MARKOV
  195 FORMAT(2X,"P",I1,"(",I1,")=",E16.10,"E**(",E16.10,"*T) + ",E16.10,MARKOV
     1       "E**(",E16.10,"*T) + ",E16.10,"E**(",E16.10,"*T)")           MARKOV
  200 CONTINUE                                                            MARKOV
  250 CONTINUE                                                            MARKOV
      RETURN                                                              MARKOV
      END                                                                 MARKOV
```

```
      SUBROUTINE MRKOVB(DELTA1,DELTA2,IXY)                                 MRKOVB
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,     MRKOVB
     1              ALPHA1,BETA1,ALPHA2,BETA2                              MRKOVB
      COMMON/EIGANL/ EV1,EV2,EV3,EL1COMP,EL2COMP,EL3COMP                   MRKOVB
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),          MRKOVB
     1               G2PWT(9,51),H2PWT(9,51)                               MRKOVB
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                         MRKOVB
C                                                                          MRKOVB
      DIMENSION CF2(9,51),CF2INT(9,51)                                     MRKOVB
C                                                                          MRKOVB
C  COMPUTE EIGENVALUES AND EIGENVECTORS FOR THIS DELTA1,DELTA2             MRKOVB
C  COMBINATION                                                             MRKOVB
      CALL EIGEN(DELTA1,DELTA2)                                            MRKOVB
C                                                                          MRKOVB
C  INITIALIZE COMMON/EIGANL/ FOR LATER USE WITH THESE PARTICULAR           MRKOVB
C  EIGENVECTORS AND EIGENVALUES                                            MRKOVB
C                                                                          MRKOVB
C  USING EIGENVALUE 1                                                      MRKOVB
      EV1=EIGWR(1)                                                         MRKOVB
      EIGL1ST=0.0                                                          MRKOVB
      X=EV1*STEP                                                           MRKOVB
      IF(X.GE.-675.84) EIGL1ST=EXP(X)                                      MRKOVB
      EL1COMP=1.0-EIGL1ST                                                  MRKOVB
C  USING EIGENVALUE 2                                                      MRKOVB
      EV2=EIGWR(2)                                                         MRKOVB
      EIGL2ST=0.0                                                          MRKOVB
      X=EV2*STEP                                                           MRKOVB
      IF(X.GE.-675.84) EIGL2ST=EXP(X)                                      MRKOVB
      EL2COMP=1.0-EIGL2ST                                                  MRKOVB
C  USING EIGENVALUE 3                                                      MRKOVB
      EV3=EIGWR(3)                                                         MRKOVB
      EIGL3ST=0.0                                                          MRKOVB
      X=EV3*STEP                                                           MRKOVB
      IF(X.GE.-675.84) EIGL3ST=EXP(X)                                      MRKOVB
      EL3COMP=1.0-EIGL3ST                                                  MRKOVB
C                                                                          MRKOVB
      A11=EIGSD(1,1,1)                                                     MRKOVB
      B11=EIGSD(1,2,1)                                                     MRKOVB
      C11=EIGSD(1,3,1)                                                     MRKOVB
C                                                                          MRKOVB
      A21=EIGSD(2,1,1)                                                     MRKOVB
      B21=EIGSD(2,2,1)                                                     MRKOVB
      C21=EIGSD(2,3,1)                                                     MRKOVB
C                                                                          MRKOVB
      A31=EIGSD(3,1,1)                                                     MRKOVB
      B31=EIGSD(3,2,1)                                                     MRKOVB
      C31=EIGSD(3,3,1)                                                     MRKOVB
C                                                                          MRKOVB
C  FOR AXY COMPUTATIONS                                                    MRKOVB
      CA=A11+A21+A31                                                       MRKOVB
      CB=B11+B21+B31                                                       MRKOVB
      CC=C11+C21+C31                                                       MRKOVB
C                                                                          MRKOVB
C  FOR APXY COMPUTATIONS                                                   MRKOVB
      CAP=BETA1*A11 + BETA2*A31                                            MRKOVB
      CBP=BETA1*B11 + BETA2*B31                                            MRKOVB
      CCP=BETA1*C11 + BETA2*C31                                            MRKOVB
C                                                                          MRKOVB
C  COMPUTE THE CF2 AND CF2INT ANALYTICAL ARRAYS REQUIRED TO COMPUTE        MRKOVB
C  THE G2WT AND H2WT WEIGHT FUNCTIONS FOR AXY AND APXY ARRAYS IN           MRKOVB
C  AORAP.                                                                  MRKOVB
```

```
C                                                                       MRKOVB
      CALL CF2ANL(CA,CB,CC,CF2,CF2INT,IXY)                              MRKOVB
      CALL G2H2WT(CF2,CF2INT,IXY,G2WT,H2WT)                             MRKOVB
C                                                                       MRKOVB
      CALL CF2ANL(CAP,CBP,CCP,CF2,CF2INT,IXY)                           MRKOVB
      CALL G2H2WT(CF2,CF2INT,IXY,G2PWT,H2PWT)                           MRKOVB
C                                                                       MRKOVB
      RETURN                                                            MRKOVB
      END                                                               MRKOVB
      SUBROUTINE CF2ANL(A,B,C,CF2,CF2INT,IXY)                           CF2ANL
      COMMON/EIGANL/ EV1,EV2,EV3,EL1COMP,EL2COMP,EL3COMP                CF2ANL
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      CF2ANL
      DIMENSION CF2(9,51),CF2INT(9,51)                                  CF2ANL
C                                                                       CF2ANL
C   THIS SUBROUTINE COMPUTES THE CAPITAL F2 AND F2P ARRAYS AND THE      CF2ANL
C   ANALYTICAL INTEGRATION OF CF2 AND CF2P FOR USE IN COMPUTINE THE     CF2ANL
C   WEIGHT ARRAYS G2WT AND H2WT FOR BOTH AXY AND APXY ARRAY COMPUTA-    CF2ANL
C   TIONS LATER IN SUBROUTINE AORAP.                                    CF2ANL
C                                                                       CF2ANL
      T=0.0                                                             CF2ANL
      DO 100 ITAU=1,ITSTPS                                              CF2ANL
      EIGL1T=0.0                                                        CF2ANL
      X=EV1*T                                                           CF2ANL
      IF(X.GE.-675.84) EIGL1T=EXP(X)                                    CF2ANL
      EIGL2T=0.0                                                        CF2ANL
      X=EV2*T                                                           CF2ANL
      IF(X.GE.-675.84) EIGL2T=EXP(X)                                    CF2ANL
      EIGL3T=0.0                                                        CF2ANL
      X=EV3*T                                                           CF2ANL
      IF(X.GE.-675.84) EIGL3T=EXP(X)                                    CF2ANL
C                                                                       CF2ANL
      CF2(IXY,ITAU)=-A*(EIGL1T/ABS(EV1))                                CF2ANL
     1              -B*(EIGL2T/ABS(EV2))                                CF2ANL
     2              -C*(EIGL3T/ABS(EV3))                                CF2ANL
C                                                                       CF2ANL
      CF2INT(IXY,ITAU)=-A*((EIGL1T*EL1COMP)/(EV1*EV1*STEP))             CF2ANL
     1                 -B*((EIGL2T*EL2COMP)/(EV2*EV2*STEP))             CF2ANL
     2                 -C*((EIGL3T*EL3COMP)/(EV3*EV3*STEP))             CF2ANL
      T=T+STEP                                                          CF2ANL
  100 CONTINUE                                                          CF2ANL
C                                                                       CF2ANL
      RETURN                                                            CF2ANL
      END                                                               CF2ANL
      SUBROUTINE G2H2WT(CF2,CF2INT,IXY,G2W,H2W)                         G2H2WT
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      G2H2WT
      DIMENSION G2W(9,51),H2W(9,51),CF2(9,51),CF2INT(9,51)              G2H2WT
C                                                                       G2H2WT
C   THIS SUBROUTINE COMPUTES ARRAYS G2WT,H2WT,G2PWT,H2PWT FOR LATER     G2H2WT
C   COMPUTATION OF AXY AND APXY ARRAYS IN SUBROUTINE AORAP.             G2H2WT
C   THESE ARRAYS ARE 0 FOR THE FIRST STEP ALTHOUGH THE 0 IS NEVER       G2H2WT
C   USED BECAUSE G2 AND H2 FUNCTIONS ARE INDEXED USING TI+1.            G2H2WT
C                                                                       G2H2WT
      ITSTM1=ITSTPS-1                                                   G2H2WT
      DO 100 ITAU=1,ITSTM1                                              G2H2WT
      ITAUP1=ITAU+1                                                     G2H2WT
      G2W(IXY,ITAUP1)=CF2INT(IXY,ITAU)-CF2(IXY,ITAU)                    G2H2WT
      H2W(IXY,ITAUP1)=CF2INT(IXY,ITAU)-CF2(IXY,ITAUP1)                  G2H2WT
  100 CONTINUE                                                          G2H2WT
C                                                                       G2H2WT
      RETURN                                                            G2H2WT
      END                                                               G2H2WT
      SUBROUTINE SUMMAT(II,JJ,KK,KURSET,IT)                             SUMMAT
```

```
C                                                                       SUMMAT
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)          SUMMAT
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,  SUMMAT
     1              ALPHA1,BETA1,ALPHA2,BETA2                           SUMMAT
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  SUMMAT
     1              G2(3,51),AT(3,51),CT(3,51)                          SUMMAT
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      SUMMAT
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),       SUMMAT
     1               G2PWT(9,51),H2PWT(9,51)                            SUMMAT
      COMMON/DEBUGC/ DBFLCD,CDYDB(51),CBXYDB(51)                        SUMMAT
C                                                                       SUMMAT
C  THE D AND B FUNCTIONS ARE NOT TIME DEPENDENT - THEY NEED ONLY        SUMMAT
C  BE COMPUTED ONCE PER VECTOR CHANGE - NCT EVERY TIME "IT" CHANGES.    SUMMAT
C  THE SEPARATE FUNCTIONS ARE DIMENSIONED TO 448 BECAUSE 448 UNIQUE     SUMMAT
C  STATE VECTORS EXIST FOR THE CURRENT MAXIMUM CASE: 15 9 5 TO 2 2 2.   SUMMAT
C  BECAUSE THERE ARE NC FNCTION DEFINITICNS AT THIS TIME FOR BMP AND    SUMMAT
C  BPM THEY ARE DUMMY PLACE HOLDERS ONLY.                               SUMMAT
C                                                                       SUMMAT
      COMMON/DBFUNCS/ DPA(448),DMA(448),DBA(448),BPPA(448),BMPA,        SUMMAT
     1   BBPA(448),BPMA,BMMA(448),BBMA(448),BPBA(448),BMBA(448),        SUMMAT
     2   BBBA(448),INDB,DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,BMB,BBB,   SUMMAT
     3   FIMO(14),FIM1MO(14),FJMO(8),FJM1MO(8),FKNO(4),FKM1NO(4),       SUMMAT
     4   FKN1(4),FKM1N1(4)                                              SUMMAT
      DIMENSION AINTGRD(9,51,3),APINTG(9,51,1)                          SUMMAT
      DIMENSION B(9),SB(9),CBXY(9,3),AXY(9),APXY(9),AXYS(3),BPRIME(9)   SUMMAT
      DIMENSION CBXYP(9)                                                SUMMAT
C  THE FOLLOWING DIMENSIONS ARE DEPENDENT UPON I VARYING FROM 0 TO NP-2 SUMMAT
C  J FROM 0 TO NM-2 AND K FROM 0 TO NB-2, WHERE THE CURRENT MAXIMUMS    SUMMAT
C  FOR NP,NM,NB ARE 15,09,05.                                           SUMMAT
      DIMENSION PXPI(14),PXPIM1(14),PXPIM2(14),                         SUMMAT
     1          PXMJ(8),PXMJM1(8),PXMJM2(8),                            SUMMAT
     2          PXBK(4),PXBKM1(4),PXBKM2(4)                             SUMMAT
C                                                                       SUMMAT
      EQUIVALENCE(BPP,B(1))                                             SUMMAT
      REAL LAMP,LAMM,LAMB,LAMBG                                         SUMMAT
C  TOTAL NUMBER OF UNIQUE STATES                                        SUMMAT
      DATA ITOTUS/448/                                                  SUMMAT
C                                                                       SUMMAT
C                                                                       SUMMAT
      IP=1                                                              SUMMAT
      IM=2                                                              SUMMAT
      IB=3                                                              SUMMAT
      I=II-1                                                            SUMMAT
      J=JJ-1                                                            SUMMAT
      K=KK-1                                                            SUMMAT
      IM1=I-1                                                           SUMMAT
      JM1=J-1                                                           SUMMAT
      KM1=K-1                                                           SUMMAT
      IM2=I-2                                                           SUMMAT
      JM2=J-2                                                           SUMMAT
      KM2=K-2
```

```
                                                                                SUMMAT
C                                                                               SUMMAT
      IF(IT.GE.4) GO TO 15                                                      SUMMAT
C   COMPUTE SUM(IS) FOR IS=IT=1,2 AND 3                                         SUMMAT
      IS=IT                                                                     SUMMAT
      GO TO 21                                                                  SUMMAT
C   FOR NON-REDUNDANT COMPUTATION PURPOSES:                                     SUMMAT
C   SHIFT SUM(2) INTO SUM(1), SHIFT SUM(3) INTO SUM(2),                         SUMMAT
C   COMPUTE SUM(3) FOR IT GREATER THAN 3.                                       SUMMAT
C   DO THE SAME MANIPULATION TO RSTSUM.                                         SUMMAT
   15 DO 20 IS=2,3                                                              SUMMAT
      SUM(IS-1)=SUM(IS)                                                         SUMMAT
      RSTSUM(IS-1)=RSTSUM(IS)                                                   SUMMAT
   20 CONTINUE                                                                  SUMMAT
      IS=3                                                                      SUMMAT
   21 CONTINUE                                                                  SUMMAT
C                                                                               SUMMAT
C                                                                               SUMMAT
      NPMIP1=NP-I+1                                                             SUMMAT
      NMMJP1=NM-J+1                                                             SUMMAT
      NBMK=NB-K                                                                 SUMMAT
      NBMKP1=NBMK+1                                                             SUMMAT
C                                                                               SUMMAT
C   COMPUTE SLAMI,SLAMJ,SLAMK                                                   SUMMAT
C                                                                               SUMMAT
      SLAMI=NPMIP1*LAMP                                                         SUMMAT
      SLAMJ=NMMJP1*LAMM                                                         SUMMAT
      SLAMK=NBMKP1*LAMB                                                         SUMMAT
C                                                                               SUMMAT
C                                                                               SUMMAT
C                                                                               SUMMAT
C   COMPUTE CBARJL                                                              SUMMAT
C                                                                               SUMMAT
      INDB=1                                                                    SUMMAT
      CDP=0.0                                                                   SUMMAT
      CDM=0.0                                                                   SUMMAT
      CDB=0.0                                                                   SUMMAT
      CBXP=0.0                                                                  SUMMAT
      CBXM=0.0                                                                  SUMMAT
      CBXB=0.0                                                                  SUMMAT
      DO 22 JL=1,3                                                              SUMMAT
      DO 22 IY=1,9                                                              SUMMAT
      CBXY(IY,JL)=0.0                                                           SUMMAT
   22 CONTINUE                                                                  SUMMAT
      DO 23 IY=1,9                                                              SUMMAT
   23 CBXYP(IY)=0.0                                                             SUMMAT
      AXYSUM=0.0                                                                SUMMAT
      AXYS(IP)=0.0                                                              SUMMAT
      AXYS(IM)=0.0                                                              SUMMAT
      AXYS(IB)=0.0                                                              SUMMAT
      APXYSUM=C.0                                                               SUMMAT
C                                                                               SUMMAT
C   COMPUTE FUNCTIONS THAT ARE NOT TIME DEPENDENT                               SUMMAT
C                                                                               SUMMAT
      IF(IT.GT.1) GO TO 35                                                      SUMMAT
C   WRITE CURRENT VECTOR TO FUNCFL                                              SUMMAT
      IF(DBFLCC.EQ.1HF) WRITE(8,499) I,J,K                                      SUMMAT
  499 FORMAT(/" D AND B FUNCTIONS FOR VECTOR (",I2,",",I2,",",I2,")")          SUMMAT
```

```
C                                                                       SUMMAT
C     COMPUTE FUNCTIONS N0,N1,M0 AND STORE IN F ARRAYS FOR LATER        SUMMAT
C     USE IN COMPUTING THE D AND B FUNCTIONS                            SUMMAT
C                                                                       SUMMAT
      DO 24 MUP1=1,II                                                   SUMMAT
      MU=MUP1-1                                                         SUMMAT
      FIM0(MUP1)=FUNCM0(NP,I,MU)                                        SUMMAT
      FIM1M0(MUP1)=FUNCM0(NP,IM1,MU)                                    SUMMAT
   24 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
      DO 26 MUPP1=1,JJ                                                  SUMMAT
      MUP=MUPP1-1                                                       SUMMAT
      FJM0(MUPP1)=FUNCM0(NM,J,MUP)                                      SUMMAT
      FJM1M0(MUPP1)=FUNCM0(NM,JM1,MUP)                                  SUMMAT
   26 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
      DO 28 NUP1=1,KK                                                   SUMMAT
      NU=NUP1-1                                                         SUMMAT
      CALL FN0N1(NB,K,NU,FN0,FN1)                                       SUMMAT
      FKN0(NUP1)=FN0                                                    SUMMAT
      FKN1(NUP1)=FN1                                                    SUMMAT
      CALL FN0N1(NB,KM1,NU,FN0,FN1)                                     SUMMAT
      FKM1N0(NUP1)=FN0                                                  SUMMAT
      FKM1N1(NUP1)=FN1                                                  SUMMAT
   28 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
C     COMPUTE D AND B FUNCTIONS AND STORE IN D AND B ARRAYS             SUMMAT
C     FOR LATER USE WITH ALL TIME STEPS.                                SUMMAT
      DO 30 NUP1=1,KK                                                   SUMMAT
      NU=NUP1-1                                                         SUMMAT
      DO 30 MUPP1=1,JJ                                                  SUMMAT
      MUP=MUPP1-1                                                       SUMMAT
      DO 30 MUP1=1,II                                                   SUMMAT
      MU=MUP1-1                                                         SUMMAT
C                                                                       SUMMAT
      CALL SDYBXY(MU,MUP,NU,I,J,K)                                      SUMMAT
C                                                                       SUMMAT
C     WRITE CONTENTS OF COMMON/DBFUNCS/ TO FUNCFL                       SUMMAT
      IF(DBFLCD.EQ.1HF) WRITE(8,500) MU,MUP,NU,DP,DM,DB,(B(IF),IF=1,9)  SUMMAT
  500 FORMAT(2X,3(I2,1X),2(6(1X,E16.10)/11X))                           SUMMAT
C                                                                       SUMMAT
      INDB=INDB+1                                                       SUMMAT
      IF(INDB.LE.ITOTUS) GO TO 30                                       SUMMAT
      PRINT*," ERROR - D AND B FUNCTIONS ARRAY OVERFLOW - ",            SUMMAT
     1        "MAX NUMBER OF UNIQUE STATES INCREASE."                   SUMMAT
      STOP                                                              SUMMAT
   30 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
   35 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
C     COMPUTE THE PX FUNCTION VALUES THAT ARE REQUIRED FOR              SUMMAT
C     THIS (I,J,K) VECTOR AND TIME "IT".                                SUMMAT
                                                                        SUMMAT
      DO 40 MUP1=1,II                                                   SUMMAT
      MU=MUP1-1                                                         SUMMAT
      PXPIM2(MUP1)=PX(MU,IM2,1,IT)                                      SUMMAT
      PXPIM1(MUP1)=PX(MU,IM1,1,IT)                                      SUMMAT
      PXPI(MUP1)=PX(MU,I,1,IT)                                          SUMMAT
   40 CONTINUE                                                          SUMMAT
```

```
C                                                                        SUMMAT
      DO 44 MUPP1=1,JJ                                                   SUMMAT
      MUP=MUPP1-1                                                        SUMMAT
      PXMJM2(MUPP1)=PX(MUP,JM2,2,IT)                                     SUMMAT
      PXMJM1(MUPP1)=PX(MUP,JM1,2,IT)                                     SUMMAT
      PXMJ(MUPP1)=PX(MUP,J,2,IT)                                         SUMMAT
   44 CONTINUE                                                           SUMMAT
C                                                                        SUMMAT
      DO 48 NUP1=1,KK                                                    SUMMAT
      NU=NUP1-1                                                          SUMMAT
      PXBKM2(NUP1)=PX(NU,KM2,3,IT)                                       SUMMAT
      PXBKM1(NUP1)=PX(NU,KM1,3,IT)                                       SUMMAT
      PXBK(NUP1)=PX(NU,K,3,IT)                                           SUMMAT
   48 CONTINUE                                                           SUMMAT
C                                                                        SUMMAT
C   BEGIN MAIN LOOPS TO SUM UP D AND B FUNCTIONS.                        SUMMAT
C                                                                        SUMMAT
      INDB=0                                                             SUMMAT
      DO 100 NUP1=1,KK                                                   SUMMAT
      NU=NUP1-1                                                          SUMMAT
      DO 100 MUPP1=1,JJ                                                  SUMMAT
      MUP=MUPP1-1                                                        SUMMAT
      DO 100 MUP1=1,II                                                   SUMMAT
      MU=MUP1-1                                                          SUMMAT
      INDB=INDB+1                                                        SUMMAT
C                                                                        SUMMAT
C   PUT D AND B FUNCTIONS IN WORKING VARIABLES                           SUMMAT
      DP=DPA(INDB)                                                       SUMMAT
      DM=DMA(INDB)                                                       SUMMAT
      DB=DBA(INDB)                                                       SUMMAT
      BPP=BPPA(INDB)                                                     SUMMAT
      BMP=BMPA                                                           SUMMAT
      BBP=BBPA(INDB)                                                     SUMMAT
      BPM=BPMA                                                           SUMMAT
      BMM=BMMA(INDB)                                                     SUMMAT
      BBM=BBMA(INDB)                                                     SUMMAT
      BPB=BPBA(INDB)                                                     SUMMAT
      BMB=BMBA(INDB)                                                     SUMMAT
      BBB=BBBA(INDB)                                                     SUMMAT
C                                                                        SUMMAT
C   PUT PX FUNCTION VALUES FOR MUP1,MUPP1 AND NUP1 IN WORKING            SUMMAT
C   VARIABLES.                                                           SUMMAT
      PPIM2=PXPIM2(MUP1)                                                 SUMMAT
      PMJM2=PXMJM2(MUPP1)                                                SUMMAT.
      PBKM2=PXBKM2(NUP1)                                                 SUMMAT
      PPIM1=PXPIM1(MUP1)                                                 SUMMAT
      PMJM1=PXMJM1(MUPP1)                                                SUMMAT
      PBKM1=PXBKM1(NUP1)                                                 SUMMAT
      PPI=PXPI(MUP1)                                                     SUMMAT
      PMJ=PXMJ(MUPP1)                                                    SUMMAT
      PBK=PXBK(NUP1)
```

```
                                                                        SUMMAT
C                                                                       SUMMAT
C     COMPUTE ALL NECESSARY NON-PRIME,PRIME AND DOUBLE PRIME            SUMMAT
C     COMBINATIONS OF THE PX FUNCTION COMPONENTS FOR USE IN COMPUTING   SUMMAT
C     THE BPRIME AND AINTGRD ARRAYS.                                    SUMMAT
C     NON-PRIME COMBINATIONS                                            SUMMAT
      PXIP=PPIM1*PMJ*PBK                                                SUMMAT
      PXJP=PPI*PMJM1*PBK                                                SUMMAT
      PXKP=PPI*PMJ*PBKM1                                                SUMMAT
C     PRIME COMBINATIONS                                                SUMMAT
      PXIJP=PPIM1*PMJM1*PBK                                             SUMMAT
      PXIKP=PPIM1*PMJ*PBKM1                                             SUMMAT
      PXJKP=PPI*PMJM1*PBKM1                                             SUMMAT
C     DOUBLE-PRIME COMBINATIONS                                         SUMMAT
      PXIDP=PPIM2*PMJ*PBK                                               SUMMAT
      PXJDP=PPI*PMJM2*PBK                                               SUMMAT
      PXKDP=PPI*PMJ*PBKM2                                               SUMMAT
C                                                                       SUMMAT
C                                                                       SUMMAT
C     MULTIPLY THE D AND B FUNCTIONS BY THE CORRESPONDING PX FUNCTION   SUMMAT
C     COMBINATION.                                                      SUMMAT
      DP=DP*PXIP                                                        SUMMAT
      DM=DM*PXJP                                                        SUMMAT
      DB=DB*PXKP                                                        SUMMAT
C                                                                       SUMMAT
      INDX=1                                                            SUMMAT
      DO 50 IY=1,3                                                      SUMMAT
      BPRIME(INDX)=B(INDX)*PXIP                                         SUMMAT
      INDX=INDX+1                                                       SUMMAT
   50 CONTINUE                                                          SUMMAT
      DO 55 IY=1,3                                                      SUMMAT
      BPRIME(INDX)=B(INDX)*PXJP                                         SUMMAT
      INDX=INDX+1                                                       SUMMAT
   55 CONTINUE                                                          SUMMAT
      DO 60 IY=1,3                                                      SUMMAT
      BPRIME(INDX)=B(INDX)*PXKP                                         SUMMAT
      INDX=INDX+1                                                       SUMMAT
   60 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
C     MULTIPLY THE B FUNCTIONS BY THE CORRESPONDING                     SUMMAT
C     GTR2 FUNCTION STORED IN ARRAY G2(IY,IT)                           SUMMAT
C     IF ALPHA1 .EQ. 0.0, GTR2 FUNCTION .EQ. 1.0, THEREFORE THERE       SUMMAT
C     IS NO NEED TO MULTIPLY BY IT.                                     SUMMAT
      IF(ALPHA1.GT.0.0) GO TO 68                                        SUMMAT
C     INITIALIZE SB ARRAY TO THE CONTENTS OF B ARRAY IF APLHA1=0.0      SUMMAT
      DO 65 IXY=1,9                                                     SUMMAT
   65 SB(IXY)=BPRIME(IXY)                                               SUMMAT
      GO TO 75                                                          SUMMAT
   68 CONTINUE                                                          SUMMAT
      INDX=1                                                            SUMMAT
      DO 70 IDUM=1,3                                                    SUMMAT
      DO 70 IX=1,3                                                      SUMMAT
      SB(INDX)=BPRIME(INDX)*G2(IX,IT)                                   SUMMAT
      INDX=INDX+1                                                       SUMMAT
   70 CONTINUE                                                          SUMMAT
   75 CONTINUE
```

```
C
C  COMPUTE THE CAPITAL D AND B FUNCTIONS AND THE SUM OF THE
C  B FUNCTIONS OVER Y.
      CDP=CDP+DP
      CDM=CDM+DM
      CDB=CDB+DB
C  SB(IY) IS THE SUM OF THE BXY'S * THE ETR2 FUNCTION
      DO 80 IY=1,3
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)
      CBXP=CBXP+SB(IY)
   80 CONTINUE
      DO 82 IY=4,6
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)
      CBXM=CBXM+SB(IY)
   82 CONTINUE
      DO 85 IY=7,9
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)
      CBXB=CBXB+SB(IY)
   85 CONTINUE
C
C  COMPUTE THE SUM OF THE BXY'S MULTIPLIED BY THE APPROPRIATE
C  PRIME OR DOUBLE-PRIME PX FUNCTION COMBINATIONS.
C  PX FUNCTION COMBINATIONS FOR JL=1=IP
      PX1=PXIDP
      PX2=PXIJP
      PX3=PXIKP
      DO 96 JL=1,3
      DO 92 IXY=1,3
      CBXY(IXY,JL)=CBXY(IXY,JL)+(B(IXY)*PX1)
      CBXY(IXY+3,JL)=CBXY(IXY+3,JL)+(B(IXY+3)*PX2)
      CBXY(IXY+6,JL)=CBXY(IXY+6,JL)+(B(IXY+6)*PX3)
   92 CONTINUE
      IF(JL.EQ.2) GO TO 94
C  PX FUNCTION COMBINATIONS FOR JL=2=IM
      PX1=PXIJP
      PX2=PXJDP
      PX3=PXJKP
      GO TO 96
C  PX FUNCTION COMBINATIONS FOR JL=3=IB
   94 PX1=PXIKP
      PX2=PXJKP
      PX3=PXKDP
   96 CONTINUE
C
  100 CONTINUE
C
C  COMPUTE CBARJL MADE  UP OF CBARI,CBARJ AND CBARK FOR FORM1
C  OF THE GENERAL FORM OF CARE3.
      CBARI=CDP+CBXP
      CBARJ=CDM+CBXM
      CBARK=CDB+CBXB
C CREATE CAPITAL DY AND BXY ARRAYS IF DBFLCD=C
      IF(DBFLCD.NE.1HC) GO TO 200
      CDYDB(IT)=CDP+CDM+CDB
      CBXYDB(IT)=CBXP+CBXM+CBXB
  200 CONTINUE
C
C  FOR PCBARJL USE FUNCTION CPSTAR
      PSTIM2=CPSTAR(IM2,NP,1,IT)
      PSTJM2=CPSTAR(JM2,NM,2,IT)
      PSTKM2=CPSTAR(KM2,NB,3,IT)
      PSTIM1=CPSTAR(IM1,NP,1,IT)
      PSTJM1=CPSTAR(JM1,NM,2,IT)
      PSTKM1=CPSTAR(KM1,NB,3,IT)
      PSTI=CPSTAR(I,NP,1,IT)
      PSTJ=CPSTAR(J,NM,2,IT)
      PSTK=CPSTAR(K,NB,3,IT)
```

```
C                                                                       SUMMAT
C   COMPUTE ALL NESESSARY NON-PRIME,PRIME AND DOUBLE PRIME              SUMMAT
C   COMBINATIONS OF THE CAPITAL P* FUNCTION - CPSTAR FOR USE IN         SUMMAT
C   COMPUTING THE AINTGRD AND APINTG ARRAYS.                            SUMMAT
C   NON-PRIME COMBINATIONS                                              SUMMAT
      PST3IP=PSTIM1*PSTJ*PSTK                                           SUMMAT
      PST3JP=PSTI*PSTJM1*PSTK                                           SUMMAT
      PST3KP=PSTI*PSTJ*PSTKM1                                           SUMMAT
C   PRIME COMBINATIONS                                                  SUMMAT
      PST3IJP=PSTIM1*PSTJM1*PSTK                                        SUMMAT
      PST3IKP=PSTIM1*PSTJ*PSTKM1                                        SUMMAT
      PST3JKP=PSTI*PSTJM1*PSTKM1                                        SUMMAT
C   DOUBLE-PRIME COMBINATIONS                                           SUMMAT
      PST3IDP=PSTIM2*PSTJ*PSTK                                          SUMMAT
      PST3JDP=PSTI*PSTJM2*PSTK                                          SUMMAT
      PST3KDP=PSTI*PSTJ*PSTKM2                                          SUMMAT
C                                                                       SUMMAT
C   FOR PERMANENT FAULT CASE, I.E. WHEN ALPHA1=0.0 AND BETA1=0.0,       SUMMAT
C   AXY AND APXY =0.0                                                   SUMMAT
      IF(ALPHA1.EQ.0.0) GO TO 450                                       SUMMAT
C                                                                       SUMMAT
C   COMPUTE AINTGRD TO BE USED IN THE CALCULATIONS OF AXY.              SUMMAT
C   COMPUTE AINTGRD(IXY,IT,JL).  BECAUSE AXY AND APXY HAVE INTEGRANDS WITSUMMAT
C   FUNCTIONS THAT ARE DEPENDENT UPON TAU AND T-TAU, THE INTEGRATION    SUMMAT
C   MUST BE PERFORMED FROM 0 TO T EACH TIME.  THEREFORE AINTGRD MUST    SUMMAT
C   RETAIN ALL "IT" COMPUTATIONS PER VECTOR.                            SUMMAT
C                                                                       SUMMAT
      G2PCOMP=1.0-G2(IP,IT)                                             SUMMAT
      G2MCOMP=1.0-G2(IM,IT)                                             SUMMAT
      G2BCOMP=1.0-G2(IB,IT)                                             SUMMAT
C   CPSTAR AND LY COMBINATIONS FOR JL=1=IP                              SUMMAT
      PST1=PST3IDP                                                      SUMMAT
      PST2=PST3IJP                                                      SUMMAT
      PST3=PST3IKP                                                      SUMMAT
      ILY=IM2                                                           SUMMAT
      JLY=JM1                                                           SUMMAT
      KLY=KM1                                                           SUMMAT
      DO 290 JL=1,3                                                     SUMMAT
      DO 275 IXY=1,3                                                    SUMMAT
      AINTGRD(IXY,IT,JL)=PST1*CBXY(IXY,JL)*LAMP*G2PCOMP                 SUMMAT
      AINTGRD(IXY+3,IT,JL)=PST2*CBXY(IXY+3,JL)*LAMM*G2MCOMP             SUMMAT
      AINTGRD(IXY+6,IT,JL)=PST3*CBXY(IXY+6,JL)*LAMB*G2BCOMP             SUMMAT
  275 CONTINUE                                                          SUMMAT
C   COMPUTE AXY FOR JL=1                                                SUMMAT
      CALL AORAP(ILY,JLY,KLY,IT,JL,G2WT,H2WT,AINTGRD,AXY)               SUMMAT
C   SUM AXY OVER X AND Y FOR JL                                         SUMMAT
      DO 280 IXAY=1,9                                                   SUMMAT
      AXYS(JL)=AXYS(JL)+AXY(IXAY)                                       SUMMAT
  280 CONTINUE                                                          SUMMAT
      IF(JL.EQ.2) GO TO 285                                             SUMMAT
C   CPSTAR AND LY COMBINATIONS FOR JL=2=IM                              SUMMAT
      PST1=PST3IJP                                                      SUMMAT
      PST2=PST3JDP                                                      SUMMAT
      PST3=PST3JKP                                                      SUMMAT
      ILY=IM1                                                           SUMMAT
      JLY=JM2                                                           SUMMAT
      KLY=KM1                                                           SUMMAT
      GO TO 290                                                         SUMMAT
C   CPSTAR AND LY COMBINATIONS FOR JL=3=IB                              SUMMAT
  285 PST1=PST3IKP                                                      SUMMAT
      PST2=PST3JKP                                                      SUMMAT
      PST3=PST3KDP                                                      SUMMAT
      ILY=IM1                                                           SUMMAT
      JLY=JM1                                                           SUMMAT
      KLY=KM2                                                           SUMMAT
  290 CONTINUE                                                          SUMMAT
```

```
C                                                                       SUMMAT
C  COMPUTE AXYSUM                                                       SUMMAT
      AXYS(IP)=AXYS(IP)*SLAMI                                           SUMMAT
      AXYS(IM)=AXYS(IM)*SLAMJ                                           SUMMAT
      AXYS(IB)=AXYS(IB)*SLAMK                                           SUMMAT
      DO 300 JL=1,3                                                     SUMMAT
  300 AXYSUM=AXYSUM+AXYS(JL)                                            SUMMAT
C                                                                       SUMMAT
C  COMPUTE APINTG TO BE USED IN THE CALCULATICN OF APXY.                SUMMAT
C  COMPUTE APINTG(IXY,IT,1)                                             SUMMAT
      DO 350 IXY=1,3                                                    SUMMAT
      APINTG(IXY,IT,1)=PST3IP*CBXYP(IXY)*LAMP*G2PCOMP                   SUMMAT
      APINTG(IXY+3,IT,1)=PST3JP*CBXYP(IXY+3)*LAMM*G2MCOMP               SUMMAT
  350 APINTG(IXY+6,IT,1)=PST3KP*CBXYP(IXY+6)*LAMB*G2BCOMP               SUMMAT
C                                                                       SUMMAT
C  COMPUTE APXY                                                         SUMMAT
      JL=1                                                              SUMMAT
      CALL AORAP(IM1,JM1,KM1,IT,JL,G2PWT,H2PWT,APINTG,APXY)             SUMMAT
C  SUM  APXY OVER X AND Y                                               SUMMAT
      DO 400 IXY=1,9                                                    SUMMAT
      APXYSUM=APXYSUM + APXY(IXY)                                       SUMMAT
  400 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
C  COMPUTE PCBARJL WHICH IS MADE UP OF PCBARI,PCBARJ,PCBARK             SUMMAT
                                                                        SUMMAT
  450 CONTINUE                                                          SUMMAT
      PCBARI=CBARI*PST3IP*SLAMI                                         SUMMAT
      PCBARJ=CBARJ*PST3JP*SLAMJ                                         SUMMAT
      PCBARK=CBARK*PST3KP*SLAMK                                         SUMMAT
C                                                                       SUMMAT
      SUM(IS)=PCBARI+PCBARJ+PCBARK                                      SUMMAT
C                                                                       SUMMAT
C                                                                       SUMMAT
C  FINISH SUMMATION COMPUTATION BY ADDINE QLT TERMS MULTIPLIED BY       SUMMAT
C  APPROPRIATE SLAM AND STORING IN ARRAY RSTSUM(IS)                     SUMMAT
C                                                                       SUMMAT
C  RETRIEVE QLT TERMS USING MAPDIM SUBROUTINE                           SUMMAT
      RSTSUM(IS)=0.0                                                    SUMMAT
      IIM1=II-1                                                         SUMMAT
      IF(IIM1.LE.0) GO TO 710                                           SUMMAT
      CALL MAPDIM(IIM1,JJ,KK,KURSET,INDX)                               SUMMAT
      RSTSUM(IS)=QLT(INDX,IT)*SLAMI                                     SUMMAT
  710 JJM1=JJ-1                                                         SUMMAT
      IF(JJM1.LE.0) GO TO 720                                           SUMMAT
      CALL MAPDIM(II,JJM1,KK,KURSET,INDX)                               SUMMAT
      RSTSUM(IS)=RSTSUM(IS)+(QLT(INDX,IT)*SLAMJ)                        SUMMAT
  720 KKM1=KK-1                                                         SUMMAT
      IF(KKM1.LE.0) GO TO 730                                           SUMMAT
      CALL MAPDIM(II,JJ,KKM1,KURSET,INDX)                               SUMMAT
      RSTSUM(IS)=RSTSUM(IS)+(QLT(INDX,IT)*SLAMK)                        SUMMAT
  730 CONTINUE                                                          SUMMAT
```

```
C                                                                       SUMMAT
C  WRITE SUM Q+PCBAR,SUM AXY, SUM APXY TO FUNCFL IF DBFLCD=S            SUMMAT
      IF(DBFLCD.NE.1HS) GO TO 900                                       SUMMAT
      IF(IT.GT.1) GO TC 800                                             SUMMAT
      WRITE(8,799)I,J,K                                                 SUMMAT
  799 FORMAT(/2X,"FOR VECTOR(I,J,K) = (",I2,",",I1,",",I1,")"/2X,       SUMMAT
     1          "IT",3X,"SUM Q+PCBAR",8X,"SUM AXY",12X,"SUM APXY"/)     SUMMAT
  800 CONTINUE                                                          SUMMAT
      QPC=RSTSUM(IS)+SUM(IS)                                            SUMMAT
      WRITE(8,899) IT,QPC,AXYSUM,APXYSUM                                SUMMAT
  899 FORMAT(2X,I2,3(3X,E16.10))                                        SUMMAT
  900 CONTINUE                                                          SUMMAT
C                                                                       SUMMAT
C  ADD APRIME SUM AND A SUM TO SUM ARRAY BEFORE INTEGRATING             SUMMAT
C  NOTE: AXYSUM AND APXYSUM ARE 0.0 IF ALPHA1 AND BETA1=0.0             SUMMAT
      SUM(IS)=SUM(IS) + AXYSUM + APXYSUM                                SUMMAT
C  SUM(IS) AND RSTSUM(IS) MUST BE DIVIDED BY EXP(-SLAML*TAU) IN SUBROU- SUMMAT
C  TINE TRAPINT AND SIMPINT BEFORE THE TOTAL IS INTEGRATED.             SUMMAT
C                                                                       SUMMAT
      RETURN                                                            SUMMAT
      END                                                               SUMMAT
      SUBROUTINE TRAPINT(SLAML,SUBINTG)                                 TRAPINT
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      TRAPINT
      SUBINTG=0.0                                                       TRAPINT
      TAU=0.0                                                           TRAPINT
      DO 60 ITAU=1,2                                                    TRAPINT
      DIVSR1=EXP(-SLAML*TAU)                                            TRAPINT
      QUO=(RSTSUM(ITAU)+SUM(ITAU))/DIVSR1                               TRAPINT
      TAU=TAU+STEP                                                      TRAPINT
      SUBINTG=SUBINTG+QUO                                               TRAPINT
   60 CONTINUE                                                          TRAPINT
      SUBINTG=STEP*SUBINTG/2.0                                          TRAPINT
      RETURN                                                            TRAPINT
      END                                                               TRAPINT
      SUBROUTINE SIMPINT(IT,SLAML,SUBINTG)                              SIMPINT
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      SIMPINT
      SUBINTG=0.0                                                       SIMPINT
      TAU=STEP*(IT-3)                                                   SIMPINT
      ITM2=IT-2                                                         SIMPINT
      IS=1                                                              SIMPINT
      DO 70 ITAU=ITM2,IT                                                SIMPINT
      DIVSR1=EXP(-SLAML*TAU)                                            SIMPINT
      TAU=TAU+STEP                                                      SIMPINT
      QUO=(RSTSUM(IS)+SUM(IS))/DIVSR1                                   SIMPINT
      IF(ITAU.NE.IT-1) GO TO 68                                         SIMPINT
      SUBINTG=SUBINTG+4.0*QUO                                           SIMPINT
      IS=IS+1                                                           SIMPINT
      GO TO 70                                                          SIMPINT
   68 SUBINTG=SUBINTG+QUO                                               SIMPINT
      IS=IS+1                                                           SIMPINT
   70 CONTINUE                                                          SIMPINT
      SUBINTG=STEP*SUBINTG/3.0                                          SIMPINT
      RETURN                                                            SIMPINT
      END                                                               SIMPINT
```

```
      SUBROUTINE AORAP(ILY,JLY,KLY,IT,JL,G2W,H2W,F1,AXY)                AORAP
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)          AORAP
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  AORAP
     1              G2(3,51),AT(3,51),CT(3,51)                          AORAP
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      AORAP
C                                                                       AORAP
C                                                                       AORAP
C  ** THIS SUBROUTINE COMPUTES EITHER AXY(I,J,K,IT) OR APXY(I,J,K,IT).  AORAP
C                                                                       AORAP
C  WHEN COMPUTING AXY(I,J,K,IT):                                        AORAP
C  G2W,H2W   -   ARE THE INTEGRATED MARKOV WEIGHTS GIVEN TO F1 FOR THE  AORAP
C                PROB. THAT SYSTEM STARTING IN STATE 1 AT TIME TAU IS   AORAP
C                STILL IN STATE 1,2 OR 3 AT TIME T.                     AORAP
C  SUMINT    -   IS THE SUM OF THE 1 STEP INTEGRALS.                    AORAP
C  AXY       -   IS THE FINAL RESULT - (NY-LY)*SUMINT.                  AORAP
C                                                                       AORAP
C  WHEN COMPUTING APXY(I,J,K,IT):                                       AORAP
C  G2W,H2W   -   ARE THE INTEGRATED MARKOV WEIGHTS GIVEN TO F1 FOR THE  AORAP
C             PROB. SYSTEM STARTING IN STATE 1 AT TAU IS                AORAP
C                STILL OR AGAIN IN STATE 1 AT TIME T  +  THE PROB.      AORAP
C                SYSTEM STARTING IN STATE 1 AT TAU IS IN STATE 3        AORAP
C                AT TIME T.                                             AORAP
C  SUMINT    -   IS THE SUM OF THE 1 STEP INTEGRALS.                    AORAP
C  AXY       -   IS THE FINAL RESULT - (NY-LY)*SUMINT.                  AORAP
C                                                                       AORAP
C  F1        -   IS THE PORTION OF THE INTEGRAND THAT IS THE CPSTAR     AORAP
C                FUNCTIONS TIMES THE CAPITAL BXY ARRAY TIMES THE APPRO- AORAP
C                PRIATE LAMBDA VALUE TIMES (1-GTR2 FUNCTION).           AORAP
C                                                                       AORAP
      DIMENSION G2W(9,51),H2W(9,51),F1(9,51,3),SUMINT(9),AXY(9)         AORAP
      IF(IT.NE.1) GO TO 75                                              AORAP
      DO 50 IXY=1,9                                                     AORAP
   50 AXY(IXY)=0.0                                                      AORAP
      RETURN                                                            AORAP
   75 CONTINUE                                                          AORAP
C                                                                       AORAP
      NT=IT-1                                                           AORAP
      DO 100 IXY=1,9                                                    AORAP
      SUMINT(IXY)=0.0                                                   AORAP
      DO 100 ITI=1,NT                                                   AORAP
      ITIP1=ITI+1                                                       AORAP
      ITMTI=(IT-ITI)+1                                                  AORAP
      ITMTIP1=(IT-ITIP1)+1                                              AORAP
      SUMINT(IXY)=SUMINT(IXY)  + ((F1(IXY,ITMTI,JL)*G2W(IXY,ITIP1))     AORAP
     1                         - (F1(IXY,ITMTIP1,JL)*H2W(IXY,ITIP1)))   AORAP
  100 CONTINUE                                                          AORAP
C                                                                       AORAP
C                                                                       AORAP
      DO 110 IXY=1,3                                                    AORAP
  110 AXY(IXY)=(NP-ILY)*SUMINT(IXY)                                     AORAP
      DO 120 IXY=4,6                                                    AORAP
  120 AXY(IXY)=(NM-JLY)*SUMINT(IXY)                                     AORAP
      DO 130 IXY=7,9                                                    AORAP
  130 AXY(IXY)=(NB-KLY)*SUMINT(IXY)                                     AORAP
C                                                                       AORAP
      RETURN                                                            AORAP
      END                                                               AORAP
```

```
      SUBROUTINE MAPDIM(II,JJ,KK,KURSET,INDX)                           MAPDIM
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)          MAPDIM
C                                                                       MAPDIM
C  THIS SUBROUTINE UNIQUELY MAPS THE VECTOR (II,JJ,KK) INTO 1 INDEX     MAPDIM
C    VALUE TO FIND THE II,JJ,KK LOCATION IN THE 2 DIMENSIONAL ARRAY QLT,MAPDIM
C    WHERE INDX IS THE FIRST SUBSCRIPT OF QLT.  EX. QLT(INDX,NPTS)      MAPDIM
C    WHERE NPTS=1,51. (II,JJ,KK MUST NOT BE REDEFINED IN THIS           MAPDIM
C    SUBROUTINE.)                                                       MAPDIM
C                                                                       MAPDIM
C  REV. 10/09/78                                                        MAPDIM
C                                                                       MAPDIM
C  THIS MAPPING IS NECESSARY BECAUSE CDC FORTRAN DOES NOT ALLOW ARRAYS  MAPDIM
C    TO HAVE MORE THAN 3 DIMENSIONS - 4 DIMENSIONS ARE REQUIRED:        MAPDIM
C    QLT(II,JJ,KK,NPTS), THUS THE NEED FOR THE II,JJ,KK MAPPING.        MAPDIM
C                                                                       MAPDIM
C                                                                       MAPDIM
C  CHECK FOR II,JJ,KK LESS THAN ONE                                     MAPDIM
      IF(II.GT.0 .AND. JJ.GT.0 .AND. KK.GT.0) GO TO 10                  MAPDIM
      PRINT 5,II,JJ,KK                                                  MAPDIM
    5 FORMAT(" ERROR - (II,JJ,KK)=(",I2,",",I1,",",I1,                  MAPDIM
     1") - VECTOR INDICIES PASSED TO MAPDIM MUST BE GREATER THAN 0.")   MAPDIM
      STOP                                                              MAPDIM
   10 CONTINUE                                                          MAPDIM
C                                                                       MAPDIM
C  DETERMINE WHICH SET II,JJ,KK WAS DEFINED IN.                         MAPDIM
      ISET=MAX0(II,JJ,KK)                                               MAPDIM
      INDX=0                                                            MAPDIM
      IF(ISET.NE.1) GO TO 25                                            MAPDIM
      INDX=1                                                            MAPDIM
      RETURN                                                            MAPDIM
   25 CONTINUE                                                          MAPDIM
C                                                                       MAPDIM
      IF(ISET.EQ.KURSET) INDX=NSET(ISET-1)                              MAPDIM
C                                                                       MAPDIM
C  INITIALIZE UNIQUELY DEFINED L"S IN ISET - NUDEF - TO 0.              MAPDIM
C  (L REPRESENTS THE UNIQUE VECTOR II,JJ,KK)                            MAPDIM
      NUDEF=0                                                           MAPDIM
      ISET1=ISET                                                        MAPDIM
      ISET2=ISET                                                        MAPDIM
      ISET3=ISET                                                        MAPDIM
C                                                                       MAPDIM
      IF(ISET1.GT.NBF) ISET1=NBF                                        MAPDIM
      IF(ISET2.GT.NMF) ISET2=NMF                                        MAPDIM
      IF(ISET3.GT.NPF) ISET3=NPF                                        MAPDIM
      DO 300 K=1,ISET1                                                  MAPDIM
      DO 200 J=1,ISET2                                                  MAPDIM
      DO 100 I=1,ISET3                                                  MAPDIM
      IF(I.LT.ISET .AND. J.LT.ISET .AND. K.LT.ISET) GO TO 100           MAPDIM
      NUDEF=NUDEF+1                                                     MAPDIM
      IF(I.EQ.II .AND. J.EQ.JJ .AND. K.EQ.KK) GO TO 400                 MAPDIM
  100 CONTINUE                                                          MAPDIM
  200 CONTINUE                                                          MAPDIM
  300 CONTINUE                                                          MAPDIM
C                                                                       MAPDIM
C  INDX OF II,JJ,KK IS TOTAL NUMBER OF UNIQUE L"S DEFINED IN THE PREV-  MAPDIM
C  IOUS SET PLUS NUMBER OF UNIQUELY DEFINED L"S IN ISET WHICH OCCUR     MAPDIM
C  BEFORE AND INCLUDE II,JJ,KK.                                         MAPDIM
  400 INDX=INDX+NUDEF                                                   MAPDIM
C                                                                       MAPDIM
      RETURN                                                            MAPDIM
      END                                                               MAPDIM
```

```
      FUNCTION FNCK(NFAC,KFAC)                                          FNCK
C       ***********************************************                 FNCK
C       *                                             *                 FNCK
C       * THIS FUNCTION COMPUTES BINOMIAL COEFFICIENTS. *               FNCK
C       *                                             *                 FNCK
C       ***********************************************                 FNCK
C     THIS FUNCTION GIVEN TWO PARAMETERS N,K                            FNCK
C     WILL COMPUTE NCK I.E. NFACT/(N-K)FACT * KFACT                     FNCK
C     THIS IS EQUIV. TO  N(N-1)...(K+1)/(N-K)FACT.                      FNCK
C     TO USE THIS FUNCTION STATE FNCK(N,K).                             FNCK
      NMINK=NFAC-KFAC                                                   FNCK
      IF (KFAC) 20,30,60                                                FNCK
   60 IF (NMINK) 20,30,40                                               FNCK
   20 WRITE(6,100)                                                      FNCK
  100 FORMAT(22H ILLEGAL COMBINATORIAL )                                FNCK
      GO TO 50                                                          FNCK
   30 FNCK=1.                                                           FNCK
      GO TO 50                                                          FNCK
   40 Y1=KFAC+1                                                         FNCK
      Z1=1.                                                             FNCK
      FNCK=1.                                                           FNCK
      DO 10 I=1,NMINK                                                   FNCK
      FNCK=FNCK*(Y1/Z1)                                                 FNCK
      Y1=Y1+1.                                                          FNCK
      Z1=Z1+1.                                                          FNCK
   10 CONTINUE                                                          FNCK
   50 CONTINUE                                                          FNCK
      RETURN                                                            FNCK
      END                                                               FNCK
      SUBROUTINE SDYBXY(MU,MUP,NU,IC,JC,KC)                             SDYBXY
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)          SDYBXY
      COMMON/DBFUNCS/ DPA(448),DMA(448),DBA(448),BPPA(448),BMPA,        SDYBXY
     1   BBPA(448),BPMA,BMMA(448),BBMA(448),BPBA(448),BMBA(448),        SDYBXY
     2   BBBA(448),INDB,DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,BMB,BBB,   SDYBXY
     3   FIMO(14),FIM1MO(14),FJMO(8),FJM1MO(8),FKNO(4),FKM1NO(4),       SDYBXY
     4   FKN1(4),FKM1N1(4)                                              SDYBXY
      LOGICAL MUSZERO                                                   SDYBXY
C                                                                       SDYBXY
C  ONLY COMPUTE THE D AND B FUNCTIONS 1 TIME PER STATE VECTOR,          SDYBXY
C  I.E. WHEN IT=1.  THE D AND B ARRAYS WILL CONTAIN THE D AND           SDYBXY
C  B FUNCTIONS FOR EACH VECTOR DEFINED BY MU,MUP,NU.  INDB              SDYBXY
C  IS THE INDEX INTO THE D AND B ARRAYS -  IT IS ALSO THE               SDYBXY
C  COUNTER OF THE MU,MUP,NU VECTORS.                                    SDYBXY
C  THE SINGLE VARIABLES DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,           SDYBXY
C  BMB,BBB ARE THE WORKING VERSIONS OF THE D AND B ARRAYS, I.E. THE     SDYBXY
C  D AND B ARRAYS ARE NEVER MODIFIED DURING I,J,K VECTOR COMPUTA-       SDYBXY
C  TIONS.  THEY CHANGE ONLY WHEN (I,J,K) CHANGES.                       SDYBXY
C                                                                       SDYBXY
C  DEFINE COMMON TERMS                                                  SDYBXY
      MUS=MU+MUP                                                        SDYBXY
      MUSZERO=.FALSE.                                                   SDYBXY
      IF(MUS.EQ.0) MUSZERO=.TRUE.                                       SDYBXY
C  1/3**(MU+MUP)                                                        SDYBXY
      PWRMUS=(1.0/3.0)**MUS                                             SDYBXY
      TWODIV3=2.0/3.0                                                   SDYBXY
      TMU=2.0*MU                                                        SDYBXY
      TMUP=2.0*MUP                                                      SDYBXY
      ICM1=IC-1                                                         SDYBXY
      JCM1=JC-1                                                         SDYBXY
      KCM1=KC-1                                                         SDYBXY
```

```
      DBCOMI=FIMO(MU+1)                                                 SDYBXY
      DBCIM1=FIM1MO(MU+1)                                               SDYBXY
      DBCOMJ=FJMO(MUP+1)                                                SDYBXY
      DBCJM1=FJM1MO(MUP+1)                                              SDYBXY
      DBOK=FKNO(NU+1)                                                   SDYBXY
      DBOKM1=FKM1NO(NU+1)                                               SDYBXY
      DB1K=FKN1(NU+1)                                                   SDYBXY
      DB1KM1=FKM1N1(NU+1)                                               SDYBXY
      NPMI=NP-ICM1                                                      SDYBXY
      NMMJ=NM-JCM1                                                      SDYBXY
      NBMK=NB-KCM1                                                      SDYBXY
C                                                                       SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION DP                                                  SDYBXY
      DP=DBCIM1*DBCOMJ*DB1K*PWRMUS*(TMU/NPMI)                           SDYBXY
      DPA(INDB)=DP                                                      SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION DM                                                  SDYBXY
      DM=DBCOMI*DBCJM1*DB1K*PWRMUS*(TMUP/NMMJ)                          SDYBXY
      DMA(INDB)=DM                                                      SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION DB                                                  SDYBXY
      DB=(3.0/NBMK)*DBCOMI*DBCOMJ*DBOKM1*((1.C-PWRMUS)-(2.0*MUS*PWRMUS))SDYBXY
      IF(.NOT.MUSZERO) DB=DB+((2.0/NBMK)*DBCOMI*DBCOMJ*DB1KM1*PWRMUS)   SDYBXY
      IF(DB.LT.0) DB=C.0                                                SDYBXY
      DBA(INDB)=DB                                                      SDYBXY
C                                                                       SDYBXY
C   CURRENTLY DEFINITIONS FOR FUNCTIONS BMP AND BPM DO NOT EXIST.        SDYBXY
      BMP=0.0                                                           SDYBXY
      BMPA=BMP                                                          SDYBXY
      BPM=0.0                                                           SDYBXY
      BPMA=BPM                                                          SDYBXY
C                                                                       SDYBXY
C     DEFINE FUNCTION BPP                                               SDYBXY
      BPP=(TMU/NPMI)*DBCIM1*DBCOMJ*DBOK                                 SDYBXY
      BPPA(INDB)=BPP                                                    SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION BMM                                                 SDYBXY
      BMM=(TMUP/NMMJ)*DBCOMI*DBCJM1*DBOK                                SDYBXY
      BMMA(INDB)=BMM                                                    SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION BPB                                                 SDYBXY
      BBBCOM=(6.C/NBMK)*PWRMUS*DBCOMI*DBCCMJ*DBOKM1                     SDYBXY
      BPB=BBBCOM*MU                                                     SDYBXY
      BPBA(INDB)=BPB                                                    SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION BMB                                                 SDYBXY
      BMB=BBBCOM*MUP                                                    SDYBXY
      BMBA(INDB)=BMB                                                    SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION BBP                                                 SDYBXY
      BPMCOM=TWODIV3*DB1K*PWRMUS                                        SDYBXY
      BBP=BPMCOM*((NPMI-TMU)/NPMI)*DBCIM1*DBCOMJ                        SDYBXY
      BBPA(INDB)=BBP                                                    SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION BBM                                                 SDYBXY
      BBM=BPMCOM*((NMMJ-TMUP)/NMMJ)*DBCOMI*DBCJM1                       SDYBXY
      BBMA(INDB)=BBM                                                    SDYBXY
C                                                                       SDYBXY
C   DEFINE FUNCTION BBB FOR MU=MUP=0, OTHERWISE BBB=0.0                 SDYBXY
      BBB=0.0                                                           SDYBXY
      IF(MUSZERO) BBB=(2.0/NBMK)*DB1KM1                                 SDYBXY
      BBBA(INDB)=BBB                                                    SDYBXY
      RETURN                                                            SDYBXY
      END                                                               SDYBXY
```

```
      FUNCTION PX(MUX,JL,NUM,IT)                                        PX
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  PX
     1              G2(3,51),AT(3,51),CT(3,51)                          PX
C                                                                       PX
C                                                                       PX
C  PX=1.0 AT TIME 0 FOR JL=MUX                                          PX
      IF(IT.NE.1 .OR. JL.NE.MUX) GO TO 50                               PX
      PX=1.0                                                            PX
      RETURN                                                            PX
   50 CONTINUE                                                          PX
C                                                                       PX
C  PX=0.0 AT TIME 0                                                     PX
      IF(IT.GT.1 .AND. JL.GE.0 .AND. JL.GE.MUX) GO TO 100               PX
      PX=0.0                                                            PX
      RETURN                                                            PX
C                                                                       PX
  100 CONTINUE                                                          PX
      BINOM=FNCK(JL,MUX)                                                PX
      APRIME=AT(NUM,IT)/(1.0-EMLAM(NUM,IT))                             PX
      APMUX=1.0                                                         PX
      IF(MUX.NE.0) APMUX=APRIME**MUX                                    PX
      APCOMP=1.0                                                        PX
      IF((JL-MUX).NE.0) APCOMP=((1.0-APRIME)**(JL-MUX))                 PX
      PX=BINOM*APMUX*APCOMP                                             PX
      RETURN                                                            PX
      END                                                               PX
      SUBROUTINE FNON1(NCON,KL,NUX,FN0,FN1)                             FNON1
      IF(NUX.GT.0) GO TO 10                                             FNON1
      FN0=1.0                                                           FNON1
      FN1=0.0                                                           FNON1
      RETURN                                                            FNON1
   10 CONTINUE                                                          FNON1
      NBMK=NCON-KL                                                      FNON1
      NBMKPNU=NBMK+NUX                                                  FNON1
      DENOMR=NBMKPNU*(NBMKPNU-1)*(NBMKPNU-2)                            FNON1
      COMTRM=NBMK*(NBMK-1)                                              FNON1
C                                                                       FNON1
      XNUMER=COMTRM*(NBMK-2)                                            FNON1
C  COMPUTE FUNCTION N0                                                  FNON1
      FN0=XNUMER/DENOMR                                                 FNON1
C                                                                       FNON1
      XNUMER=3.0*COMTRM*NUX                                             FNON1
C  COMPUTE FUNCTION N1                                                  FNON1
      FN1=XNUMER/DENOMR                                                 FNON1
      RETURN                                                            FNON1
      END                                                               FNON1
      FUNCTION FUNCMO(NCON,JL,MUX)                                      FUNCMO
      IF(MUX.GT.1) GO TO 10                                             FUNCMO
      FUNCMO=1.0                                                        FUNCMO
      RETURN                                                            FUNCMO
   10 CONTINUE                                                          FUNCMO
      NXMLPMU=NCON-JL+MUX                                               FUNCMO
      XNUMER=1.0                                                        FUNCMO
      DENOMR=1.0                                                        FUNCMO
      MUM1=MUX-1                                                        FUNCMO
      DO 100 MULT=1,MUM1                                                FUNCMO
      XNUMER=(NXMLPMU-(3*MULT))*XNUMER                                  FUNCMO
      DENOMR=(NXMLPMU-MULT)*DENOMR                                      FUNCMO
  100 CONTINUE                                                          FUNCMO
      FUNCMO=XNUMER/DENOMR                                              FUNCMO
      RETURN                                                            FUNCMO
      END                                                               FUNCMO
```

```
      FUNCTION GTR2(XLAM,DELTA2,NUM,IT)                                 GTR2
C                                        REV.  1/26/79                  GTR2
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,  GTR2
     1              ALPHA1,BETA1,ALPHA2,BETA2                           GTR2
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  GTR2
     1              G2(3,51),AT(3,51),CT(3,51)                          GTR2
      REAL LAMP,LAMM,LAMB,LAMBG                                         GTR2
C                                                                       GTR2
C  AT TIME 0.0, GTR2=1.0; IT ALSO EQUALS 1.0 IF ALPHA1=0.0 AT ANY TIME  GTR2
      IF(ALPHA1.EQ.0.0) GO TO 2                                         GTR2
      IF(IT.GT.1) GO TO 5                                               GTR2
    2 GTR2=1.0                                                          GTR2
      RETURN                                                            GTR2
C                                                                       GTR2
C                                                                       GTR2
C  COMPUTE TERMS THAT ARE USED MORE THAN CNCE.                          GTR2
C                                                                       GTR2
    5 CONTINUE                                                          GTR2
      XLAM1=XLAM12(1,DELTA2)                                            GTR2
      XLAM2=XLAM12(2,DELTA2)                                            GTR2
C                                                                       GTR2
      EL=EMLAM(NUM,IT)                                                  GTR2
      EL1=EMLAM1(NUM,IT)                                                GTR2
      EL2=EMLAM2(NUM,IT)                                                GTR2
C                                                                       GTR2
      XLML1=XLAM-XLAM1                                                  GTR2
      XLML2=XLAM-XLAM2                                                  GTR2
      XL1ML2=XLAM1-XLAM2                                                GTR2
C                                                                       GTR2
      XNUMER=(ALPHA1*XLML1*EL2)-(ALPHA1*XLML2*EL1)+(ALPHA1*XL1ML2*EL)   GTR2
      DENOMR=(XLML2*(DELTA2-XLAM2)*EL1) - (XLKL1*(DELTA2-XLAM1)*EL2)    GTR2
     1       + (XL1ML2*(XLAM1+XLAM2-XLAM-DELTA2)*EL)                    GTR2
      IF(DENOMR.NE.0.0) GO TO 20                                        GTR2
      PRINT 999,XLML2,DELTA2,XLAM2,EL1,XLML1,XLAM1,EL2,XL1ML2,XLAM,EL   GTR2
  999 FORMAT(" DENOMR IN FUNCTION GTR2, IS 0.0"/                        GTR2
     1" XLML2",13X,"DELTA2",12X,"XLAM2",13X,"EL1",15X,"XLML1",13X,      GTR2
     2"XLAM1",13X,"EL2"/1X,6(E16.10,2X)/                                GTR2
     3"XL1ML2",12X,"XLAM",14X,"EL"/1X,3(E16.10,2X))                     GTR2
      STOP                                                              GTR2
   20 TERM=XNUMER/DENOMR                                                GTR2
C                                                                       GTR2
      GTR2=1.0-TERM                                                     GTR2
      RETURN                                                            GTR2
      END                                                               GTR2
      FUNCTION AFUNC(XLAM,DELTA,NUM,IT)                                 AFUNC
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,CELTAP,DELTAM,DELTAB,DELTABG,  AFUNC
     1              ALPHA1,BETA1,ALPHA2,BETA2                           AFUNC
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  AFUNC
     1              G2(3,51),AT(3,51),CT(3,51)                          AFUNC
      REAL LAMP,LAMM,LAMB,LAMBG                                         AFUNC
C                                                                       AFUNC
      EML=EMLAM(NUM,IT)                                                 AFUNC
C                                                                       AFUNC
C  CODE FOR BUS RATES                                                   AFUNC
      IF(NUM.NE.3) GO TO 10                                             AFUNC
      AFUNC=(XLAM/(DELTA-XLAM)) * (EML-EMDEL(NUM,IT))                   AFUNC
      GO TO 20                                                          AFUNC
C                                                                       AFUNC
C  CODE FOR PROCESSOR AND MEMORY RATES                                  AFUNC
   10 CONTINUE                                                          AFUNC
      TERM1=(LAMBG/(DELTABG-XLAM)) * (EML-EMDEL(4,IT))                  AFUNC
      TERM2=((XLAM-LAMBG)/(DELTA-XLAM)) * (EML-EMDEL(NUM,IT))           AFUNC
      AFUNC=TERM1+TERM2                                                 AFUNC
C                                                                       AFUNC
   20 CONTINUE                                                          AFUNC
      RETURN                                                            AFUNC
      END                                                               AFUNC
```

```
      SUBROUTINE AEFUNCS(XLAM,DELTA,NUM,IT,ATR,BTR)                     ABFUNCS
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  ABFUNCS
     1              G2(3,51),AT(3,51),CT(3,51)                          ABFUNCS
C                                                                       ABFUNCS
C  COMPUTE TERMS THAT ARE USED MORE THAN ONCE                           ABFUNCS
C                                                                       ABFUNCS
      XLAM1=XLAM12(1,DELTA)                                             ABFUNCS
      XLAM2=XLAM12(2,DELTA)                                             ABFUNCS
C                                                                       ABFUNCS
      DENOMR1=(XLAM1-XLAM2)*(XLAM-XLAM1)                                ABFUNCS
      DENOMR2=(XLAM1-XLAM2)*(XLAM-XLAM2)                                ABFUNCS
      DENOMR3=(XLAM-XLAM1)*(XLAM-XLAM2)                                 ABFUNCS
C                                                                       ABFUNCS
      IF(DENOMR1.NE.0.0 .AND. DENOMR2.NE.0.0 .AND. DENOMR3.NE.0.0)      ABFUNCS
     1   GO TO 10                                                       ABFUNCS
      PRINT 99,XLAM1,XLAM2,XLAM                                         ABFUNCS
   99 FORMAT(" DENOMR IN FUNCTION ATR IS 0.0"/                          ABFUNCS
     1" XLAM1",13X,"XLAM2",13X,"XLAM"/1X,3(E16.10,2X))                  ABFUNCS
      STOP                                                              ABFUNCS
C                                                                       ABFUNCS
   10 TERM1=((XLAM*(DELTA-XLAM2))/DENOMR1) * EMLAM1(NUM,IT)             ABFUNCS
      TERM2=((XLAM*(DELTA-XLAM1))/DENOMR2) * EMLAM2(NUM,IT)             ABFUNCS
      TERM3=((XLAM*(XLAM1+XLAM2-DELTA-XLAM))/DENOMR3) * EMLAM(NUM,IT)   ABFUNCS
      ATR=TERM1-TERM2+TERM3                                             ABFUNCS
      TERM4=((XLAM1*XLAM2-XLAM*DELTA)/DENOMR3) * EMLAM(NUM,IT)          ABFUNCS
      BTR=1.0-(TERM1-TERM2+TERM4)                                       ABFUNCS
      RETURN                                                            ABFUNCS
      END                                                               ABFUNCS
      FUNCTION XLAM12(PORM,DELTA)                                       XLAM12
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,  XLAM12
     1              ALPHA1,BETA1,ALPHA2,BETA2                           XLAM12
C                                                                       XLAM12
C  PARAMETER PORM(PLUS OR MINUS) DETERMINES WHETHER THE 2 TERMS OF      XLAM12
C  THIS FUNCTION SHOULD BE ADDED OR SUBTRACTED.                         XLAM12
C                                                                       XLAM12
      REAL LAMP,LAMM,LAMB,LAMBG                                         XLAM12
      INTEGER PORM                                                      XLAM12
C                                                                       XLAM12
      SUBTRM=ALPHA1+DELTA+BETA1                                         XLAM12
      TERM1=0.5*SUBTRM                                                  XLAM12
      SUBTRM2=SUBTRM*SUBTRM                                             XLAM12
      TERM2=0.5*SQRT(SUBTRM2-(4.0*BETA1*DELTA))                         XLAM12
C                                                                       XLAM12
      IF(PORM.NE.1) GO TO 10                                            XLAM12
      XLAM12=TERM1+TERM2                                                XLAM12
      GO TO 30                                                          XLAM12
   10 IF(PORM.NE.2) GO TO 20                                            XLAM12
      XLAM12=TERM1-TERM2                                                XLAM12
      GO TO 30                                                          XLAM12
   20 PRINT*," ERROR - PORM PARAMETER MUST EQUAL EITHER 1 OR 2 ",PORM   XLAM12
   30 RETURN                                                            XLAM12
      END                                                               XLAM12
```

```
      FUNCTION PSTAR(JL,NCON,NUM,IT)                                    PSTAR
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  PSTAR
     1              G2(3,51),AT(3,51),CT(3,51)                          PSTAR
      IF(JL) 10,20,30                                                   PSTAR
C                                                                       PSTAR
   10 PSTAR=0.0                                                         PSTAR
      RETURN                                                            PSTAR
C                                                                       PSTAR
   20 PSTAR=1.0                                                         PSTAR
      RETURN                                                            PSTAR
C                                                                       PSTAR
   30 ECOMP=1.0 - EMLAM(NUM,IT)                                         PSTAR
      ECOMPP=ECOMP                                                      PSTAR
      IF(JL.EQ.1) GO TO 50                                              PSTAR
   35 DO 40 IP=2,JL                                                     PSTAR
      ECOMPP=ECOMP*ECOMPP                                               PSTAR
   40 CONTINUE                                                          PSTAR
C                                                                       PSTAR
   50 PSTAR=FNCK(NCON,JL)*ECOMPP                                        PSTAR
      RETURN                                                            PSTAR
      END                                                               PSTAR
      FUNCTION CPSTAR(JL,NCON,NUM,IT)                                   CPSTAR
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  CPSTAR
     1              G2(3,51),AT(3,51),CT(3,51)                          CPSTAR
      PART=PSTAR(JL,NCON,NUM,IT)                                        CPSTAR
C  COMPUTE EXP RAISED TO THE POWER -LAM*T,NCON-JL TIMES                 CPSTAR
      EMLT=EMLAM(NUM,IT)                                                CPSTAR
      EMLTP=EMLT                                                        CPSTAR
      NMULTS=NCON-JL                                                    CPSTAR
      IF(NMULTS.LE.1) GO TO 20                                          CPSTAR
      DO 10 ITIMES=2,NMULTS                                             CPSTAR
      EMLTP=EMLT*EMLTP                                                  CPSTAR
   10 CONTINUE                                                          CPSTAR
C                                                                       CPSTAR
   20 CPSTAR=PART*EMLTP                                                 CPSTAR
      RETURN                                                            CPSTAR
      END                                                               CPSTAR
      SUBROUTINE SUMRMA(II,JJ,KK,KURSET,IT)                             SUMRMA
C                                                                       SUMRMA
      COMMON/CONFIG/ NF,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)          SUMRMA
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,  SUMRMA
     1              ALPHA1,BETA1,ALPHA2,BETA2                           SUMRMA
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),  SUMRMA
     1              G2(3,51),AT(3,51),CT(3,51)                          SUMRMA
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                      SUMRMA
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),       SUMRMA
     1              G2PWT(9,51),H2PWT(9,51)                             SUMRMA
      COMMON/DEBUGC/ DBFLCD,CDYDB(51),CBXYDB(51)                        SUMRMA
C                                                                       SUMRMA
C  THE D AND B FUNCTIONS ARE NOT TIME DEPENDENT - THEY NEED ONLY        SUMRMA
C  BE COMPUTED ONCE PER VECTOR CHANGE - NOT EVERY TIME "IT" CHANGES.    SUMRMA
C  THE SEPARATE FUNCTIONS ARE DIMENSIONED TO 448 BECAUSE 448 UNIQUE     SUMRMA
C  STATE VECTORS EXIST FOR THE CURRENT MAXIMUM CASE: 15 9 5 TO 2 2 2.   SUMRMA
C  BECAUSE THERE ARE NO FNCTION DEFINITIONS AT THIS TIME FOR BMP AND    SUMRMA
C  BPM THEY ARE DUMMY PLACE HOLDERS ONLY.                               SUMRMA
C                                                                       SUMRMA
      COMMON/DBFUNCS/ DPA(448),DMA(448),DBA(448),BPPA(448),BMPA,        SUMRMA
     1   BBPA(448),BPMA,BMMA(448),BBMA(448),BPBA(448),BMBA(448),        SUMRMA
     2   BBBA(448),INDB,DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,BMB,BBB,   SUMRMA
     3   FIMO(14),FIM1MO(14),FJMC(8),FJM1MO(8),FKNO(4),FKM1NO(4),       SUMRMA
     4   FKN1(4),FKM1N1(4)                                              SUMRMA
      DIMENSION AINTGRD(9,51,3),APINTG(9,51,1)                          SUMRMA
      DIMENSION B(9),SB(9),CBXY(9,3),AXY(9),APXY(9),AXYS(3),BPRIME(9)   SUMRMA
      DIMENSION CBXYP(9)                                                SUMRMA
```

```
C                                                                       SUMRMA
      EQUIVALENCE(BPP,B(1))                                             SUMRMA
      REAL LAMP,LAMM,LAMB,LAMBG                                         SUMRMA
C   TOTAL NUMBER OF UNIQUE STATES                                       SUMRMA
      DATA ITOTUS/448/                                                  SUMRMA
C                                                                       SUMRMA
C                                                                       SUMRMA
      IP=1                                                              SUMRMA
      IM=2                                                              SUMRMA
      IB=3                                                              SUMRMA
      I=II-1                                                            SUMRMA
      J=JJ-1                                                            SUMRMA
      K=KK-1                                                            SUMRMA
      IM1=I-1                                                           SUMRMA
      JM1=J-1                                                           SUMRMA
      KM1=K-1                                                           SUMRMA
      IM2=I-2                                                           SUMRMA
      JM2=J-2                                                           SUMRMA
      KM2=K-2                                                           SUMRMA
C                                                                       SUMRMA
      IF(IT.GE.4) GO TO 15                                              SUMRMA
C   COMPUTE SUM(IS) FOR IS=IT=1,2 AND 3                                 SUMRMA
      IS=IT                                                             SUMRMA
      GO TO 21                                                          SUMRMA
C   FOR NON-REDUNDANT COMPUTATION PURPOSES:                             SUMRMA
C   SHIFT SUM(2) INTO SUM(1), SHIFT SUM(3) INTO SUM(2),                 SUMRMA
C   COMPUTE SUM(3) FOR IT GREATER THAN 3.                               SUMRMA
C   DO THE SAME MANIPULATION TO RSTSUM.                                 SUMRMA
   15 DO 20 IS=2,3                                                      SUMRMA
      SUM(IS-1)=SUM(IS)                                                 SUMRMA
      RSTSUM(IS-1)=RSTSUM(IS)                                           SUMRMA
   20 CONTINUE                                                          SUMRMA
      IS=3                                                              SUMRMA
   21 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C                                                                       SUMRMA
      NPMIP1=NP-I+1                                                     SUMRMA
      NMMJP1=NM-J+1                                                     SUMRMA
      NBMK=NB-K                                                         SUMRMA
      NBMKP1=NBMK+1                                                     SUMRMA
C                                                                       SUMRMA
C   COMPUTE SLAMI,SLAMJ,SLAMK                                           SUMRMA
C                                                                       SUMRMA
      SLAMI=NPMIP1*LAMP                                                 SUMRMA
      SLAMJ=NMMJP1*LAMM                                                 SUMRMA
      SLAMK=NBMKP1*LAMB                                                 SUMRMA
C                                                                       SUMRMA
C                                                                       SUMRMA
C                                                                       SUMRMA
C   COMPUTE CBARJL                                                      SUMRMA
C                                                                       SUMRMA
      INDB=1                                                            SUMRMA
      CDP=0.0                                                           SUMRMA
      CDM=0.0                                                           SUMRMA
      CDB=0.0                                                           SUMRMA
      CBXP=0.0                                                          SUMRMA
      CBXM=0.0                                                          SUMRMA
      CBXB=0.0                                                          SUMRMA
      DO 22 JL=1,3                                                      SUMRMA
      DO 22 IY=1,9                                                      SUMRMA
      CBXY(IY,JL)=0.0                                                   SUMRMA
   22 CONTINUE                                                          SUMRMA
```

```
      DO 23 IY=1,9                                                      SUMRMA
   23 CBXYP(IY)=0.0                                                     SUMRMA
      AXYSUM=0.0                                                        SUMRMA
      AXYS(IP)=0.0                                                      SUMRMA
      AXYS(IM)=0.0                                                      SUMRMA
      AXYS(IB)=0.0                                                      SUMRMA
      APXYSUM=0.0                                                       SUMRMA
C                                                                       SUMRMA
C  COMPUTE FUNCTIONS THAT ARE NOT TIME DEFENDENT                        SUMRMA
C                                                                       SUMRMA
      IF(IT.GT.1) GO TO 35                                              SUMRMA
C  WRITE CURRENT VECTOR TO FUNCFL                                       SUMRMA
      IF(DBFLCD.EQ.1HF) WRITE(8,499) I,J,K                              SUMRMA
  499 FORMAT(/" D AND B FUNCTIONS FOR VECTOR (",I2,",",I2,",",I2,")")   SUMRMA
C                                                                       SUMRMA
C  COMPUTE FUNCTIONS N0,N1,M0 AND STORE IN F ARRAYS FOR LATER           SUMRMA
C  USE IN COMPUTING THE D AND B FUNCTIONS                               SUMRMA
C                                                                       SUMRMA
      DO 24 MUP1=1,II                                                   SUMRMA
      MU=MUP1-1                                                         SUMRMA
      FIMO(MUP1)=FUNCMO(NP,I,MU)                                        SUMRMA
      FIM1MO(MUP1)=FUNCMO(NP,IM1,MU)                                    SUMRMA
   24 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
      DO 26 MUPP1=1,JJ                                                  SUMRMA
      MUP=MUPP1-1                                                       SUMRMA
      FJMO(MUPP1)=FUNCMO(NM,J,MUP)                                      SUMRMA
      FJM1MO(MUPP1)=FUNCMO(NM,JM1,MUP)                                  SUMRMA
   26 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
      DO 28 NUP1=1,KK                                                   SUMRMA
      NU=NUP1-1                                                         SUMRMA
      CALL FNON1(NB,K,NU,FN0,FN1)                                       SUMRMA
      FKN0(NUP1)=FN0                                                    SUMRMA
      FKN1(NUP1)=FN1                                                    SUMRMA
      CALL FNON1(NB,KM1,NU,FN0,FN1)                                     SUMRMA
      FKM1N0(NUP1)=FN0                                                  SUMRMA
      FKM1N1(NUP1)=FN1                                                  SUMRMA
   28 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C  COMPUTE D AND B FUNCTIONS AND STORE IN D AND B ARRAYS                SUMRMA
C  FOR LATER USE WITH ALL TIME STEPS.                                   SUMRMA
      DO 30 NUP1=1,KK                                                   SUMRMA
      NU=NUP1-1                                                         SUMRMA
      DO 30 MUPP1=1,JJ                                                  SUMRMA
      MUP=MUPP1-1                                                       SUMRMA
      DO 30 MUP1=1,II                                                   SUMRMA
      MU=MUP1-1                                                         SUMRMA
C                                                                       SUMRMA
      CALL SDBRA(MU,MUP,NU,I,J,K)                                       SUMRMA
C                                                                       SUMRMA
C  WRITE CONTENTS OF COMMON/DBFUNCS/ TO FUNCFL                          SUMRMA
      IF(DBFLCD.EQ.1HF) WRITE(8,500) MU,MUP,NU,DP,DM,DB,(B(IF),IF=1,9)  SUMRMA
  500 FORMAT(2X,3(I2,1X),2(6(1X,E16.10)/11X))                           SUMRMA
C                                                                       SUMRMA
      INDB=INDB+1                                                       SUMRMA
      IF(INDB.LE.ITOTUS) GO TO 30                                       SUMRMA
      PRINT*," ERROR - D AND B FUNCTIONS ARRAY OVERFLOW - ",            SUMRMA
     1       "MAX NUMBER OF UNIQUE STATES INCREASE."                    SUMRMA
      STOP                                                              SUMRMA
   30 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
   35 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
```

```
C                                                                       SUMRMA
C  BEGIN MAIN LOOPS TO SUM UP D AND B FUNCTIONS.                         SUMRMA
C                                                                       SUMRMA
      INDB=0                                                            SUMRMA
      DO 100 NUP1=1,KK                                                  SUMRMA
      NU=NUP1-1                                                         SUMRMA
      DO 100 MUPP1=1,JJ                                                 SUMRMA
      MUP=MUPP1-1                                                       SUMRMA
      DO 100 MUP1=1,II                                                  SUMRMA
      MU=MUP1-1                                                         SUMRMA
      INDB=INDB+1                                                       SUMRMA
C                                                                       SUMRMA
C  PUT D AND B FUNCTIONS IN WORKING VARIABLES                           SUMRMA
      DP=DPA(INDB)                                                      SUMRMA
      DM=DMA(INDB)                                                      SUMRMA
      DB=DBA(INDB)                                                      SUMRMA
      BPP=BPPA(INDB)                                                    SUMRMA
      BMP=BMPA                                                          SUMRMA
      BBP=BBPA(INDB)                                                    SUMRMA
      BPM=BPMA                                                          SUMRMA
      BMM=BMMA(INDB)                                                    SUMRMA
      BBM=BBMA(INDB)                                                    SUMRMA
      BPB=BPBA(INDB)                                                    SUMRMA
      BMB=BMBA(INDB)                                                    SUMRMA
      BBB=BBBA(INDB)                                                    SUMRMA
C                                                                       SUMRMA
C  INITIALIZE PCOND VARIABLES TO 0.0                                    SUMRMA
      PCIP=0.0                                                          SUMRMA
      PCJP=0.0                                                          SUMRMA
      PCKP=0.0                                                          SUMRMA
      PCIJP=0.0                                                         SUMRMA
      PCIKP=0.0                                                         SUMRMA
      PCJKP=0.0                                                         SUMRMA
      PCIDP=0.0                                                         SUMRMA
      PCJDP=0.0                                                         SUMRMA
      PCKDP=0.0                                                         SUMRMA
      IF((MU+MUP+NU).GT.2) GO TO 40                                     SUMRMA
C                                                                       SUMRMA
C  COMPUTE ALL NECESSARY NON-PRIME,PRIME AND DOUBLE PRIME               SUMRMA
C  COMBINATIONS OF THE PCOND FUNCTION FOR USE IN COMPUTING              SUMRMA
C  THE BPRIME AND AINTGRD ARRAYS FOR MU+MUP+NU.LE.2.                    SUMRMA
C  NON-PRIME COMBINATIONS                                               SUMRMA
      PCIP=PCOND(MU,MUP,NU,IM1,J,K,IT)                                  SUMRMA
      PCJP=PCOND(MU,MUP,NU,I,JM1,K,IT)                                  SUMRMA
      PCKP=PCOND(MU,MUP,NU,I,J,KM1,IT)                                  SUMRMA
C  PRIME COMBINATIONS                                                   SUMRMA
      PCIJP=PCOND(MU,MUP,NU,IM1,JM1,K,IT)                               SUMRMA
      PCIKP=PCOND(MU,MUP,NU,IM1,J,KM1,IT)                               SUMRMA
      PCJKP=PCOND(MU,MUP,NU,I,JM1,KM1,IT)                               SUMRMA
C  DOUBLE-PRIME COMBINATIONS                                            SUMRMA
      PCIDP=PCOND(MU,MUP,NU,IM2,J,K,IT)                                 SUMRMA
      PCJDP=PCOND(MU,MUP,NU,I,JM2,K,IT)                                 SUMRMA
      PCKDP=PCOND(MU,MUP,NU,I,J,KM2,IT)                                 SUMRMA
   40 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C                                                                       SUMRMA
C  MULTIPLY THE D AND B FUNCTIONS BY THE CORRESPONDING PCOND FUNCTION   SUMRMA
C  COMBINATION.                                                         SUMRMA
      DP=DP*PCIP                                                        SUMRMA
      DM=DM*PCJP                                                        SUMRMA
      DB=DB*PCKP                                                        SUMRMA
```

```
C                                                                       SUMRMA
      INDX=1                                                            SUMRMA
      DO 50 IY=1,3                                                      SUMRMA
      BPRIME(INDX)=B(INDX)*PCIP                                         SUMRMA
      INDX=INDX+1                                                       SUMRMA
   50 CONTINUE                                                          SUMRMA
      DO 55 IY=1,3                                                      SUMRMA
      BPRIME(INDX)=B(INDX)*PCJP                                         SUMRMA
      INDX=INDX+1                                                       SUMRMA
   55 CONTINUE                                                          SUMRMA
      DO 60 IY=1,3                                                      SUMRMA
      BPRIME(INDX)=B(INDX)*PCKP                                         SUMRMA
      INDX=INDX+1                                                       SUMRMA
   60 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C     MULTIPLY THE B FUNCTIONS BY THE CORRESPONDING                     SUMRMA
C     GTR2 FUNCTION STORED IN ARRAY G2(IY,IT)                           SUMRMA
C     IF ALPHA1 .EQ. 0.0, GTR2 FUNCTION .EQ. 1.0, THEREFORE THERE       SUMRMA
C     IS NO NEED TO MULTIPLY BY IT.                                     SUMRMA
      IF(ALPHA1.GT.0.0) GO TO 68                                        SUMRMA
C     INITIALIZE SB ARRAY TO THE CONTENTS OF B ARRAY IF APLHA1=0.0      SUMRMA
      DO 65 IXY=1,9                                                     SUMRMA
   65 SB(IXY)=BPRIME(IXY)                                               SUMRMA
      GO TO 75                                                          SUMRMA
   68 CONTINUE                                                          SUMRMA
      INDX=1                                                            SUMRMA
      DO 70 IDUM=1,3                                                    SUMRMA
      DO 70 IX=1,3                                                      SUMRMA
      SB(INDX)=BPRIME(INDX)*G2(IX,IT)                                   SUMRMA
      INDX=INDX+1                                                       SUMRMA
   70 CONTINUE                                                          SUMRMA
   75 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C     COMPUTE THE CAPITAL D AND B FUNCTIONS AND THE SUM OF THE          SUMRMA
C     B FUNCTIONS OVER X.                                               SUMRMA
      CDP=CDP+DP                                                        SUMRMA
      CDM=CDM+DM                                                        SUMRMA
      CDB=CDB+DB                                                        SUMRMA
C     SB(IY) IS THE SUM OF THE BXY'S * THE GTR2 FUNCTION                SUMRMA
      DO 80 IY=1,3                                                      SUMRMA
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)                                    SUMRMA
      CBXP=CBXP+SB(IY)                                                  SUMRMA
   80 CONTINUE                                                          SUMRMA
      DO 82 IY=4,6                                                      SUMRMA
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)                                    SUMRMA
      CBXM=CBXM+SB(IY)                                                  SUMRMA
   82 CONTINUE                                                          SUMRMA
      DO 85 IY=7,9                                                      SUMRMA
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)                                    SUMRMA
      CBXB=CBXB+SB(IY)                                                  SUMRMA
   85 CONTINUE                                                          SUMRMA
```

```
C                                                                       SUMRMA
C   COMPUTE THE SUM OF THE BXY'S MULTIPLIED BY THE APPROPRIATE          SUMRMA
C   PRIME OR DOUBLE-PRIME PCOND FUNCTION COMBINATIONS.                  SUMRMA
C   PCOND FUNCTION COMBINATIONS FOR JL=1=IP                             SUMRMA
      PC1=PCIDP                                                         SUMRMA
      PC2=PCIJP                                                         SUMRMA
      PC3=PCIKP                                                         SUMRMA
      DO 96 JL=1,3                                                      SUMRMA
      DO 92 IXY=1,3                                                     SUMRMA
      CBXY(IXY,JL)=CBXY(IXY,JL)+(B(IXY)*PC1)                            SUMRMA
      CBXY(IXY+3,JL)=CBXY(IXY+3,JL)+(B(IXY+3)*PC2)                      SUMRMA
      CBXY(IXY+6,JL)=CBXY(IXY+6,JL)+(B(IXY+6)*PC3)                      SUMRMA
   92 CONTINUE                                                          SUMRMA
      IF(JL.EQ.2) GO TO 94                                              SUMRMA
C   PCOND FUNCTION COMBINATIONS FOR JL=2=IM                             SUMRMA
      PC1=PCIJP                                                         SUMRMA
      PC2=PCJDP                                                         SUMRMA
      PC3=PCJKP                                                         SUMRMA
      GO TO 96                                                          SUMRMA
C   PCOND FUNCTION COMBINATIONS FOR JL=3=IB                             SUMRMA
   94 PC1=PCIKP                                                         SUMRMA
      PC2=PCJKP                                                         SUMRMA
      PC3=PCKDP                                                         SUMRMA
   96 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
  100 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C   COMPUTE CBARJL MADE  UP OF CBARI,CBARJ AND CBARK FOR FORM1          SUMRMA
C   OF THE GENERAL FORM OF CARE3.                                       SUMRMA
      CBARI=CDP+CBXP                                                    SUMRMA
      CBARJ=CDM+CBXM                                                    SUMRMA
      CBARK=CDB+CBXB                                                    SUMRMA
C  CREATE CAPITAL DY AND BXY ARRAYS IF DBFLCD=C                         SUMRMA
      IF(DBFLCD.NE.1HC) GO TO 200                                       SUMRMA
      CDYDB(IT)=CDP+CDM+CDB                                             SUMRMA
      CBXYDB(IT)=CBXP+CBXM+CBXB                                         SUMRMA
  200 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C    FOR PCBARJL USE FUNCTION CPSTAR                                    SUMRMA
      PSTIM2=CPSTAR(IM2,NP,1,IT)                                        SUMRMA
      PSTJM2=CPSTAR(JM2,NM,2,IT)                                        SUMRMA
      PSTKM2=CPSTAR(KM2,NB,3,IT)                                        SUMRMA
      PSTIM1=CPSTAR(IM1,NP,1,IT)                                        SUMRMA
      PSTJM1=CPSTAR(JM1,NM,2,IT)                                        SUMRMA
      PSTKM1=CPSTAR(KM1,NB,3,IT)                                        SUMRMA
      PSTI=CPSTAR(I,NP,1,IT)                                            SUMRMA
      PSTJ=CPSTAR(J,NM,2,IT)                                            SUMRMA
      PSTK=CPSTAR(K,NB,3,IT)                                            SUMRMA
C                                                                       SUMRMA
C    COMPUTE ALL NESESSARY NON-PRIME,PRIME AND DOUBLE PRIME             SUMRMA
C    COMBINATIONS OF THE CAPITAL P* FUNCTION - CPSTAR FOR USE IN        SUMRMA
C    COMPUTING THE AINTGRD AND APINTG ARRAYS.                           SUMRMA
C    NON-PRIME COMBINATIONS                                             SUMRMA
      PST3IP=PSTIM1*PSTJ*PSTK                                           SUMRMA
      PST3JP=PSTI*PSTJM1*PSTK                                           SUMRMA
      PST3KP=PSTI*PSTJ*PSTKM1                                           SUMRMA
C    PRIME COMBINATIONS                                                 SUMRMA
      PST3IJP=PSTIM1*PSTJM1*PSTK                                        SUMRMA
      PST3IKP=PSTIM1*PSTJ*PSTKM1                                        SUMRMA
      PST3JKP=PSTI*PSTJM1*PSTKM1                                        SUMRMA
C    DOUBLE-PRIME COMBINATIONS                                          SUMRMA
      PST3IDP=PSTIM2*PSTJ*PSTK                                          SUMRMA
      PST3JDP=PSTI*PSTJM2*PSTK                                          SUMRMA
      PST3KDP=PSTI*PSTJ*PSTKM2                                          SUMRMA
```

```
C                                                                       SUMRMA
C   FOR PERMANENT FAULT CASE, I.E. WHEN ALPHA1=0.0 AND BETA1=0.0,      SUMRMA
C   AXY AND APXY =0.0                                                   SUMRMA
      IF(ALPHA1.EQ.0.0) GO TO 450                                       SUMRMA
C                                                                       SUMRMA
C   COMPUTE AINTGRD TO BE USED IN THE CALCULATIONS OF AXY.             SUMRMA
C   COMPUTE AINTGRD(IXY,IT,JL).  BECAUSE AXY AND APXY HAVE INTEGRANDS WITSUMRMA
C   FUNCTIONS THAT ARE DEPENDENT UPON TAU AND T-TAU, THE INTEGRATION   SUMRMA
C   MUST BE PERFORMED FROM 0 TO T EACH TIME.  THEREFORE AINTGRD MUST   SUMRMA
C   RETAIN ALL "IT" COMPUTATIONS PER VECTOR.                            SUMRMA
C                                                                       SUMRMA
      G2PCOMP=1.0-G2(IP,IT)                                             SUMRMA
      G2MCOMP=1.0-G2(IM,IT)                                             SUMRMA
      G2BCOMP=1.0-G2(IB,IT)                                             SUMRMA
C   CPSTAR AND LY COMBINATIONS FOR JL=1=IP                              SUMRMA
      PST1=PST3IDP                                                      SUMRMA
      PST2=PST3IJP                                                      SUMRMA
      PST3=PST3IKP                                                      SUMRMA
      ILY=IM2                                                           SUMRMA
      JLY=JM1                                                           SUMRMA
      KLY=KM1                                                           SUMRMA
      DO 290 JL=1,3                                                     SUMRMA
      DO 275 IXY=1,3                                                    SUMRMA
      AINTGRD(IXY,IT,JL)=PST1*CBXY(IXY,JL)*LAMP*G2PCOMP                 SUMRMA
      AINTGRD(IXY+3,IT,JL)=PST2*CBXY(IXY+3,JL)*LAMM*G2MCOMP             SUMRMA
      AINTGRD(IXY+6,IT,JL)=PST3*CBXY(IXY+6,JL)*LAMB*G2BCOMP             SUMRMA
  275 CONTINUE                                                          SUMRMA
C   COMPUTE AXY FOR JL=1                                                SUMRMA
      CALL AORAP(ILY,JLY,KLY,IT,JL,G2WT,H2WT,AINTGRD,AXY)               SUMRMA
C   SUM AXY OVER X AND Y FOR JL                                         SUMRMA
      DO 280 IXAY=1,9                                                   SUMRMA
      AXYS(JL)=AXYS(JL)+AXY(IXAY)                                       SUMRMA
  280 CONTINUE                                                          SUMRMA
      IF(JL.EQ.2) GO TO 285                                             SUMRMA
C   CPSTAR AND LY COMBINATIONS FOR JL=2=IM                              SUMRMA
      PST1=PST3IJP                                                      SUMRMA
      PST2=PST3JDP                                                      SUMRMA
      PST3=PST3JKP                                                      SUMRMA
      ILY=IM1                                                           SUMRMA
      JLY=JM2                                                           SUMRMA
      KLY=KM1                                                           SUMRMA
      GO TO 290                                                         SUMRMA
C   CPSTAR AND LY COMBINATIONS FOR JL=3=IB                              SUMRMA
  285 PST1=PST3IKP                                                      SUMRMA
      PST2=PST3JKP                                                      SUMRMA
      PST3=PST3KDP                                                      SUMRMA
      ILY=IM1                                                           SUMRMA
      JLY=JM1                                                           SUMRMA
      KLY=KM2                                                           SUMRMA
  290 CONTINUE                                                          SUMRMA
C                                                                       SUMRMA
C   COMPUTE AXYSUM                                                      SUMRMA
      AXYS(IP)=AXYS(IP)*SLAMI                                           SUMRMA
      AXYS(IM)=AXYS(IM)*SLAMJ                                           SUMRMA
      AXYS(IB)=AXYS(IB)*SLAMK                                           SUMRMA
      DO 300 JL=1,3                                                     SUMRMA
  300 AXYSUM=AXYSUM+AXYS(JL)                                            SUMRMA
C                                                                       SUMRMA
C   COMPUTE APINTG TO BE USED IN THE CALCULATION OF APXY.              SUMRMA
C   COMPUTE APINTG(IXY,IT,1)                                            SUMRMA
      DO 350 IXY=1,3                                                    SUMRMA
      APINTG(IXY,IT,1)=PST3IP*CBXYP(IXY)*LAMP*G2PCOMP                   SUMRMA
      APINTG(IXY+3,IT,1)=PST3JP*CBXYP(IXY+3)*LAMM*G2MCOMP               SUMRMA
  350 APINTG(IXY+6,IT,1)=PST3KP*CBXYP(IXY+6)*LAMB*G2BCOMP               SUMRMA
```

```
C                                                                       SUMRMA
C   COMPUTE APXY                                                        SUMRMA
       JL=1                                                             SUMRMA
       CALL AORAP(IM1,JM1,KM1,IT,JL,G2PWT,H2PWT,APINTG,APXY)            SUMRMA
C   SUM  APXY OVER X AND Y                                              SUMRMA
       DO 400 IXY=1,9                                                   SUMRMA
       APXYSUM=APXYSUM + APXY(IXY)                                      SUMRMA
  400  CONTINUE                                                         SUMRMA
C                                                                       SUMRMA
C   COMPUTE PCBARJL WHICH IS MADE UP OF PCBARI,PCBARJ,PCBARK            SUMRMA
                                                                        SUMRMA
  450  CONTINUE                                                         SUMRMA
       PCBARI=CBARI*PST3IP*SLAMI                                        SUMRMA
       PCBARJ=CBARJ*PST3JP*SLAMJ                                        SUMRMA
       PCBARK=CBARK*PST3KP*SLAMK                                        SUMRMA
C                                                                       SUMRMA
       SUM(IS)=PCBARI+PCBARJ+PCBARK                                     SUMRMA
C                                                                       SUMRMA
C                                                                       SUMRMA
C   FINISH SUMMATION COMPUTATION BY ADDING QLT TERMS MULTIPLIED BY      SUMRMA
C   APPROPRIATE SLAM AND STORING IN ARRAY RSTSUM(IS)                    SUMRMA
C                                                                       SUMRMA
C   RETRIEVE QLT TERMS USING MAPDIM SUBROUTINE                          SUMRMA
       RSTSUM(IS)=0.0                                                   SUMRMA
       IIM1=II-1                                                        SUMRMA
       IF(IIM1.LE.0) GO TO 710                                          SUMRMA
       CALL MAPDIM(IIM1,JJ,KK,KURSET,INDX)                              SUMRMA
       RSTSUM(IS)=QLT(INDX,IT)*SLAMI                                    SUMRMA
  710  JJM1=JJ-1                                                        SUMRMA
       IF(JJM1.LE.0) GO TO 720                                          SUMRMA
       CALL MAPDIM(II,JJM1,KK,KURSET,INDX)                              SUMRMA
       RSTSUM(IS)=RSTSUM(IS)+(QLT(INDX,IT)*SLAMJ)                       SUMRMA
  720  KKM1=KK-1                                                        SUMRMA
       IF(KKM1.LE.0) GO TO 730                                          SUMRMA
       CALL MAPDIM(II,JJ,KKM1,KURSET,INDX)                              SUMRMA
       RSTSUM(IS)=RSTSUM(IS)+(QLT(INDX,IT)*SLAMK)                       SUMRMA
  730  CONTINUE                                                         SUMRMA
C                                                                       SUMRMA
C   WRITE SUM Q+PCBAR,SUM AXY, SUM APXY TO FUNCFL IF DBFLCD=S           SUMRMA
       IF(DBFLCD.NE.1HS) GO TO 900                                      SUMRMA
       IF(IT.GT.1) GO TO 800                                            SUMRMA
       WRITE(8,799)I,J,K                                                SUMRMA
  799  FORMAT(/2X,"FOR VECTOR(I,J,K) = (",I2,",",I1,",",I1,")"/2X,      SUMRMA
      1          "IT",3X,"SUM Q+PCBAR",8X,"SUM AXY",12X,"SUM APXY"/)    SUMRMA
  800  CONTINUE                                                         SUMRMA
       QPC=RSTSUM(IS)+SUM(IS)                                           SUMRMA
       WRITE(8,899) IT,QPC,AXYSUM,APXYSUM                               SUMRMA
  899  FORMAT(2X,I2,3(3X,E16.10))                                       SUMRMA
  900  CONTINUE                                                         SUMRMA
C                                                                       SUMRMA
C   ADD APRIME SUM AND A SUM TO SUM ARRAY BEFORE INTEGRATING            SUMRMA
C   NOTE: AXYSUM AND APXYSUM ARE 0.0 IF ALPHA1 AND BETA1=0.0            SUMRMA
       SUM(IS)=SUM(IS) + AXYSUM + APXYSUM                               SUMRMA
C   SUM(IS) AND RSTSUM(IS) MUST BE DIVIDED BY EXP(-SLAML*TAU) IN SUBROU- SUMRMA
C   TINE TRAPINT AND SIMPINT BEFORE THE TOTAL IS INTEGRATED.            SUMRMA
C                                                                       SUMRMA
       RETURN                                                           SUMRMA
       END                                                              SUMRMA
```

```
      SUBROUTINE SUMRMB(II,JJ,KK,KURSET,IT)                               SUMRMB
C                                                                         SUMRMB
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)            SUMRMB
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,    SUMRMB
     1              ALPHA1,BETA1,ALPHA2,BETA2                             SUMRMB
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),    SUMRMB
     1              G2(3,51),AT(3,51),CT(3,51)                            SUMRMB
      COMMON/INTGRAT/ ITSTPS,STEP,SUM(3),RSTSUM(3)                        SUMRMB
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),         SUMRMB
     1              G2PWT(9,51),H2PWT(9,51)                               SUMRMB
      COMMON/DEBUGC/ DBFLCD,CDYDB(51),CBXYDB(51)                          SUMRMB
C                                                                         SUMRMB
C   THE D AND B FUNCTIONS ARE NOT TIME DEPENDENT - THEY NEED ONLY         SUMRMB
C   BE COMPUTED ONCE PER VECTOR CHANGE - NCT EVERY TIME "IT" CHANGES.     SUMRMB
C   THE SEPARATE FUNCTIONS ARE DIMENSIONED TO 448 BECAUSE 448 UNIQUE      SUMRMB
C   STATE VECTORS EXIST FOR THE CURRENT MAXIMUM CASE: 15 9 5 TO 2 2 2.    SUMRMB
C   BECAUSE THERE ARE NO FNCTION DEFINITIONS AT THIS TIME FOR BMP AND     SUMRMB
C   BPM THEY ARE DUMMY PLACE HOLDERS ONLY.                                SUMRMB
C                                                                         SUMRMB
      COMMON/DBFUNCS/ DPA(448),DMA(448),DEA(448),BPPA(448),BMPA,          SUMRMB
     1   BBPA(448),BPMA,BMMA(448),BBMA(448),BPBA(448),BMBA(448),          SUMRMB
     2   BBBA(448),INDB,DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,BMB,BBB,     SUMRMB
     3   FIMO(14),FIM1MO(14),FJMC(8),FJM1MC(8),FKNO(4),FKM1NO(4),         SUMRMB
     4   FKN1(4),FKM1N1(4)                                                SUMRMB
      DIMENSION APINTG(9,51,1)                                            SUMRMB
      DIMENSION B(9),SB(9),APXY(9),BPRIME(9)                              SUMRMB
      DIMENSION CBXYP(9)                                                  SUMRMB
C                                                                         SUMRMB
      EQUIVALENCE(BPP,B(1))                                               SUMRMB
      REAL LAMP,LAMM,LAMB,LAMBG                                           SUMRMB
C   TOTAL NUMBER OF UNIQUE STATES                                         SUMRMB
      DATA ITOTUS/448/                                                    SUMRMB
C                                                                         SUMRMB
C                                                                         SUMRMB
      IP=1                                                                SUMRMB
      IM=2                                                                SUMRMB
      IB=3                                                                SUMRMB
      I=II-1                                                              SUMRMB
      J=JJ-1                                                              SUMRMB
      K=KK-1                                                              SUMRMB
      IM1=I-1                                                             SUMRMB
      JM1=J-1                                                             SUMRMB
      KM1=K-1                                                             SUMRMB
C                                                                         SUMRMB
      IF(IT.GE.4) GO TO 15                                                SUMRMB
C   COMPUTE SUM(IS) FOR IS=IT=1,2 AND 3                                   SUMRMB
      IS=IT                                                               SUMRMB
      GO TO 21                                                            SUMRMB
C   FOR NON-REDUNDANT COMPUTATION PURPOSES:                               SUMRMB
C   SHIFT SUM(2) INTO SUM(1), SHIFT SUM(3) INTO SUM(2),                   SUMRMB
C   COMPUTE SUM(3) FOR IT GREATER THAN 3.                                 SUMRMB
C   DO THE SAME MANIPULATION TO RSTSUM.                                   SUMRMB
   15 DO 20 IS=2,3                                                        SUMRMB
      SUM(IS-1)=SUM(IS)                                                   SUMRMB
      RSTSUM(IS-1)=RSTSUM(IS)                                             SUMRMB
   20 CONTINUE                                                            SUMRMB
      IS=3                                                                SUMRMB
   21 CONTINUE                                                            SUMRMB
C                                                                         SUMRMB
```

```
C                                                                     SUMRMB
      NPMIP1=NP-I+1                                                   SUMRMB
      NMMJP1=NM-J+1                                                   SUMRMB
      NBMK=NB-K                                                       SUMRMB
      NBMKP1=NBMK+1                                                   SUMRMB
C                                                                     SUMRMB
C  COMPUTE SLAMI,SLAMJ,SLAMK                                          SUMRMB
C                                                                     SUMRMB
      SLAMI=NPMIP1*LAMP                                               SUMRMB
      SLAMJ=NMMJP1*LAMM                                               SUMRMB
      SLAMK=NBMKP1*LAMB                                               SUMRMB
C                                                                     SUMRMB
C                                                                     SUMRMB
C                                                                     SUMRMB
C  COMPUTE CBARJL                                                     SUMRMB
C                                                                     SUMRMB
      INDB=1                                                          SUMRMB
      CDP=0.0                                                         SUMRMB
      CDM=0.0                                                         SUMRMB
      CDB=0.0                                                         SUMRMB
      CBXP=0.0                                                        SUMRMB
      CBXM=0.0                                                        SUMRMB
      CBXB=0.0                                                        SUMRMB
      DO 23 IY=1,9                                                    SUMRMB
   23 CBXYP(IY)=0.0                                                   SUMRMB
      APXYSUM=0.0                                                     SUMRMB
C                                                                     SUMRMB
C  COMPUTE FUNCTIONS THAT ARE NOT TIME DEPENDENT                      SUMRMB
C                                                                     SUMRMB
      IF(IT.GT.1) GO TO 35                                            SUMRMB
C  WRITE CURRENT VECTOR TO FUNCFL                                     SUMRMB
      IF(DBFLCC.EQ.1HF) WRITE(8,499) I,J,K                            SUMRMB
  499 FORMAT(/" D AND B FUNCTICNS FOR VECTOR (",I2,",",I2,",",I2,")") SUMRMB
C                                                                     SUMRMB
C  COMPUTE FUNCTIONS NO,N1,MO AND STORE IN F ARRAYS FOR LATER         SUMRMB
C  USE IN COMPUTING THE D AND B FUNCTIONS                             SUMRMB
C                                                                     SUMRMB
      DO 24 MUP1=1,II                                                 SUMRMB
      MU=MUP1-1                                                       SUMRMB
      FIMO(MUP1)=FUNCMO(NP,I,MU)                                      SUMRMB
      FIM1MO(MUP1)=FUNCMO(NP,IM1,MU)                                  SUMRMB
   24 CONTINUE                                                        SUMRMB
C                                                                     SUMRMB
      DO 26 MUPP1=1,JJ                                                SUMRMB
      MUP=MUPP1-1                                                     SUMRMB
      FJMO(MUPP1)=FUNCMO(NM,J,MUP)                                    SUMRMB
      FJM1MO(MUPP1)=FUNCMO(NM,JM1,MUP)                                SUMRMB
   26 CONTINUE                                                        SUMRMB
C                                                                     SUMRMB
      DO 28 NUP1=1,KK                                                 SUMRMB
      NU=NUP1-1                                                       SUMRMB
      CALL FNON1(NB,K,NU,FNO,FN1)                                     SUMRMB
      FKNO(NUP1)=FNO                                                  SUMRMB
      FKN1(NUP1)=FN1                                                  SUMRMB
      CALL FNON1(NB,KM1,NU,FNO,FN1)                                   SUMRMB
      FKM1NO(NUP1)=FNO                                                SUMRMB
      FKM1N1(NUP1)=FN1                                                SUMRMB
   28 CONTINUE                                                        SUMRMB
```

```
C                                                                       SUMRMB
C   COMPUTE D AND B FUNCTIONS AND STORE IN D AND B ARRAYS              SUMRMB
C   FOR LATER USE WITH ALL TIME STEPS.                                  SUMRMB
      DO 30 NUP1=1,KK                                                   SUMRMB
      NU=NUP1-1                                                         SUMRMB
      DO 30 MUPP1=1,JJ                                                  SUMRMB
      MUP=MUPP1-1                                                       SUMRMB
      DO 30 MUP1=1,II                                                   SUMRMB
      MU=MUP1-1                                                         SUMRMB
C                                                                       SUMRMB
      CALL SDBRB(MU,MUP,NU,I,J,K)                                       SUMRMB
C                                                                       SUMRMB
C   WRITE CONTENTS OF COMMON/DBFUNCS/ TO FUNCFL                         SUMRMB
      IF(DBFLCD.EQ.1HF) WRITE(8,500) MU,MUP,NU,DP,DM,DB,(B(IF),IF=1,9)  SUMRMB
  500 FORMAT(2X,3(I2,1X),2(6(1X,E16.10)/11X))                           SUMRMB
C                                                                       SUMRMB
      INDB=INDB+1                                                       SUMRMB
      IF(INDB.LE.ITOTUS) GO TO 30                                       SUMRMB
      PRINT*," ERROR - D AND B FUNCTIONS ARRAY OVERFLOW - ",            SUMRMB
     1       "MAX NUMBER OF UNIQUE STATES INCREASE."                    SUMRMB
      STOP                                                              SUMRMB
   30 CONTINUE                                                          SUMRMB
C                                                                       SUMRMB
   35 CONTINUE                                                          SUMRMB
C                                                                       SUMRMB
C                                                                       SUMRMB
C   BEGIN MAIN LOOPS TO SUM UP D AND B FUNCTIONS.                       SUMRMB
C                                                                       SUMRMB
      INDB=0                                                            SUMRMB
      DO 100 NUP1=1,KK                                                  SUMRMB
      NU=NUP1-1                                                         SUMRMB
      DO 100 MUPP1=1,JJ                                                 SUMRMB
      MUP=MUPP1-1                                                       SUMRMB
      DO 100 MUP1=1,II                                                  SUMRMB
      MU=MUP1-1                                                         SUMRMB
      INDB=INDB+1                                                       SUMRMB
C                                                                       SUMRMB
C   PUT D AND B FUNCTIONS IN WORKING VARIABLES                          SUMRMB
      DP=DPA(INDB)                                                      SUMRMB
      DM=DMA(INDB)                                                      SUMRMB
      DB=DBA(INDB)                                                      SUMRMB
      BPP=BPPA(INDB)                                                    SUMRMB
      BMP=BMPA                                                          SUMRMB
      BBP=BBPA(INDB)                                                    SUMRMB
      BPM=BPMA                                                          SUMRMB
      BMM=BMMA(INDB)                                                    SUMRMB
      BBM=BBMA(INDB)                                                    SUMRMB
      BPB=BPBA(INDB)                                                    SUMRMB
      BMB=BMBA(INDB)                                                    SUMRMB
      BBB=BBBA(INDB)                                                    SUMRMB
C                                                                       SUMRMB
C   COMPUTE PCOND FUNCTIONS IF (MU+MUP+NU.LE.2)                         SUMRMB
      PCIP=0.0                                                          SUMRMB
      PCJP=0.0                                                          SUMRMB
      PCKP=0.0                                                          SUMRMB
      IF((MU+MUP+NU).GT.2) GO TO 40                                     SUMRMB
      PCIP=PCOND(MU,MUP,NU,IM1,J,K,IT)                                  SUMRMB
      PCJP=PCOND(MU,MUP,NU,I,JM1,K,IT)                                  SUMRMB
      PCKP=PCOND(MU,MUP,NU,I,J,KM1,IT)                                  SUMRMB
   40 CONTINUE                                                          SUMRMB
C                                                                       SUMRMB
C                                                                       SUMRMB
```

```
C                                                                        SUMRMB
C   MULTIPLY THE D AND B FUNCTIONS BY THE CORRESPONDING PX FUNCTION     SUMRMB
C   COMBINATION.                                                         SUMRMB
      DP=DP*PCIP                                                         SUMRMB
      DM=DM*PCJP                                                         SUMRMB
      DB=DB*PCKP                                                         SUMRMB
C                                                                        SUMRMB
      INDX=1                                                             SUMRMB
      DO 50 IY=1,3                                                       SUMRMB
      BPRIME(INDX)=B(INDX)*PCIP                                          SUMRMB
      INDX=INDX+1                                                        SUMRMB
   50 CONTINUE                                                           SUMRMB
      DO 55 IY=1,3                                                       SUMRMB
      BPRIME(INDX)=B(INDX)*PCJP                                          SUMRMB
      INDX=INDX+1                                                        SUMRMB
   55 CONTINUE                                                           SUMRMB
      DO 60 IY=1,3                                                       SUMRMB
      BPRIME(INDX)=B(INDX)*PCKP                                          SUMRMB
      INDX=INDX+1                                                        SUMRMB
   60 CONTINUE                                                           SUMRMB
C                                                                        SUMRMB
C   MULTIPLY THE B FUNCTIONS BY THE CORRESPONDING                        SUMRMB
C   GTR2 FUNCTION STORED IN ARRAY G2(IY,IT)                              SUMRMB
C   IF ALPHA1 .EQ. 0.0, GTR2 FUNCTION .EQ. 1.0, THEREFORE THERE          SUMRMB
C   IS NO NEED TO MULTIPLY BY IT.                                        SUMRMB
      IF(ALPHA1.GT.0.0) GO TO 68                                         SUMRMB
C   INITIALIZE SB ARRAY TO THE CONTENTS OF B ARRAY IF APLHA1=0.0         SUMRMB
      DO 65 IXY=1,9                                                      SUMRMB
   65 SB(IXY)=BPRIME(IXY)                                                SUMRMB
      GO TO 75                                                           SUMRMB
   68 CONTINUE                                                           SUMRMB
      INDX=1                                                             SUMRMB
      DO 70 IDUM=1,3                                                     SUMRMB
      DO 70 IX=1,3                                                       SUMRMB
      SB(INDX)=BPRIME(INDX)*G2(IX,IT)                                    SUMRMB
      INDX=INDX+1                                                        SUMRMB
   70 CONTINUE                                                           SUMRMB
   75 CONTINUE                                                           SUMRMB
C                                                                        SUMRMB
C   COMPUTE THE CAPITAL D AND B FUNCTIONS AND THE SUM OF THE             SUMRMB
C   B FUNCTIONS OVER X.                                                  SUMRMB
      CDP=CDP+DP                                                         SUMRMB
      CDM=CDM+DM                                                         SUMRMB
      CDB=CDB+DB                                                         SUMRMB
C   SB(IY) IS THE SUM OF THE BXY'S * THE GTR2 FUNCTION                   SUMRMB
      DO 80 IY=1,3                                                       SUMRMB
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)                                     SUMRMB
      CBXP=CBXP+SB(IY)                                                   SUMRMB
   80 CONTINUE                                                           SUMRMB
      DO 82 IY=4,6                                                       SUMRMB
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)                                     SUMRMB
      CBXM=CBXM+SB(IY)                                                   SUMRMB
   82 CONTINUE                                                           SUMRMB
      DO 85 IY=7,9                                                       SUMRMB
      CBXYP(IY)=CBXYP(IY)+BPRIME(IY)                                     SUMRMB
      CBXB=CBXB+SB(IY)                                                   SUMRMB
   85 CONTINUE                                                           SUMRMB
C                                                                        SUMRMB
  100 CONTINUE                                                           SUMRMB
```

```
C                                                                       SUMRMB
C   COMPUTE CBARJL MADE  UP OF CBARI,CBARJ AND CBARK FOR FORM1          SUMRMB
C   OF THE GENERAL FORM OF CARE3.                                       SUMRMB
      CBARI=CDP+CBXP                                                    SUMRMB
      CBARJ=CDM+CBXM                                                    SUMRMB
      CBARK=CDB+CBXB                                                    SUMRMB
C CREATE CAPITAL DY AND BXY ARRAYS IF DBFLCD=C                          SUMRMB
      IF(DBFLCD.NE.1HC) GO TO 200                                       SUMRMB
      CDYDB(IT)=CDP+CDM+CDB                                             SUMRMB
      CBXYDB(IT)=CBXP+CBXM+CBXB                                         SUMRMB
  200 CONTINUE                                                          SUMRMB
C                                                                       SUMRMB
C   FOR PCBARJL USE FUNCTION CPSTAR                                     SUMRMB
      PSTIM1=CPSTAR(IM1,NP,1,IT)                                        SUMRMB
      PSTJM1=CPSTAR(JM1,NM,2,IT)                                        SUMRMB
      PSTKM1=CPSTAR(KM1,NB,3,IT)                                        SUMRMB
      PSTI=CPSTAR(I,NP,1,IT)                                            SUMRMB
      PSTJ=CPSTAR(J,NM,2,IT)                                            SUMRMB
      PSTK=CPSTAR(K,NB,3,IT)                                            SUMRMB
C                                                                       SUMRMB
C   COMPUTE ALL NESESSARY                                               SUMRMB
C   COMBINATIONS OF THE CAPITAL P* FUNCTION - CPSTAR FOR USE IN         SUMRMB
C   COMPUTING THE APINTG ARRAY.                                         SUMRMB
      PST3IP=PSTIM1*PSTJ*PSTK                                           SUMRMB
      PST3JP=PSTI*PSTJM1*PSTK                                           SUMRMB
      PST3KP=PSTI*PSTJ*PSTKM1                                           SUMRMB
C                                                                       SUMRMB
C   FOR PERMANENT FAULT CASE, I.E. WHEN ALPHA1=0.0 AND BETA1=0.0,       SUMRMB
C   AXY AND APXY =0.0                                                   SUMRMB
      IF(ALPHA1.EQ.0.0) GO TO 450                                       SUMRMB
C                                                                       SUMRMB
C   COMPUTE APINTG TO BE USED IN THE CALCULATION OF APXY.               SUMRMB
C   COMPUTE APINTG(IXY,IT,1)                                            SUMRMB
      G2PCOMP=1.0-G2(IP,IT)                                             SUMRMB
      G2MCOMP=1.0-G2(IM,IT)                                             SUMRMB
      G2BCOMP=1.0-G2(IB,IT)                                             SUMRMB
      DO 350 IXY=1,3                                                    SUMRMB
      APINTG(IXY,IT,1)=PST3IP*CBXYP(IXY)*LAMP*G2PCOMP                   SUMRMB
      APINTG(IXY+3,IT,1)=PST3JP*CBXYP(IXY+3)*LAMM*G2MCOMP               SUMRMB
  350 APINTG(IXY+6,IT,1)=PST3KP*CBXYP(IXY+6)*LAMB*G2BCOMP               SUMRMB
C                                                                       SUMRMB
C   COMPUTE APXY                                                        SUMRMB
      JL=1                                                              SUMRMB
      CALL AORAP(IM1,JM1,KM1,IT,JL,G2PWT,H2PWT,APINTG,APXY)             SUMRMB
C   SUM  APXY OVER X AND Y                                              SUMRMB
      DO 400 IXY=1,9                                                    SUMRMB
      APXYSUM=APXYSUM+APXY(IXY)                                         SUMRMB
  400 CONTINUE                                                          SUMRMB
C                                                                       SUMRMB
C   COMPUTE PCBARJL WHICH IS MADE UP OF PCBARI,PCBARJ,PCBARK            SUMRMB
                                                                        SUMRMB
  450 CONTINUE                                                          SUMRMB
      PCBARI=CBARI*PST3IP*SLAMI                                         SUMRMB
      PCBARJ=CBARJ*PST3JP*SLAMJ                                         SUMRMB
      PCBARK=CBARK*PST3KP*SLAMK                                         SUMRMB
C                                                                       SUMRMB
      SUM(IS)=PCBARI+PCBARJ+PCBARK                                      SUMRMB
C                                                                       SUMRMB
C                                                                       SUMRMB
C   FINISH SUMMATION COMPUTATION BY ADDING QLT TERMS MULTIPLIED BY      SUMRMB
C   APPROPRIATE SLAM AND STORING IN ARRAY RSTSUM(IS)                    SUMRMB
```

```
C                                                                       SUMRMB
C   RETRIEVE QLT TERMS USING MAFDIM SUBROUTINE                          SUMRMB
      RSTSUM(IS)=0.0                                                    SUMRMB
      IIM1=II-1                                                         SUMRMB
      IF(IIM1.LE.0) GO TO 710                                           SUMRMB
      CALL MAPDIM(IIM1,JJ,KK,KURSET,INDX)                               SUMRMB
      RSTSUM(IS)=QLT(INDX,IT)*SLAMI                                     SUMRMB
  710 JJM1=JJ-1                                                         SUMRMB
      IF(JJM1.LE.0) GO TO 720                                           SUMRMB
      CALL MAPDIM(II,JJM1,KK,KURSET,INDX)                               SUMRMB
      RSTSUM(IS)=RSTSUM(IS)+(QLT(INDX,IT)*SLAMJ)                        SUMRMB
  720 KKM1=KK-1                                                         SUMRMB
      IF(KKM1.LE.0) GO TO 730                                           SUMRMB
      CALL MAPDIM(II,JJ,KKM1,KURSET,INDX)                               SUMRMB
      RSTSUM(IS)=RSTSUM(IS)+(QLT(INDX,IT)*SLAMK)                        SUMRMB
  730 CONTINUE                                                          SUMRMB
C                                                                       SUMRMB
C   WRITE SUM Q+PCBAR,SUM AXY, SUM APXY TO FUNCFL IF DBFLCD=S           SUMRMB
      IF(DBFLCD.NE.1HS) GO TO 900                                       SUMRMB
      IF(IT.GT.1) GO TO 800                                             SUMRMB
      WRITE(8,799)I,J,K                                                 SUMRMB
  799 FORMAT(/2X,"FOR VECTOR(I,J,K) = (",I2,",",I1,",",I1,")"/2X,       SUMRMB
     1            "IT",3X,"SUM Q+PCBAR",8X,"SUM APXY"/)                 SUMRMB
  800 CONTINUE                                                          SUMRMB
      QPC=RSTSUM(IS)+SUM(IS)                                            SUMRMB
      WRITE(8,899) IT,QPC,APXYSUM                                       SUMRMB
  899 FORMAT(2X,I2,2(3X,E16.10))                                        SUMRMB
  900 CONTINUE                                                          SUMRMB
C                                                                       SUMRMB
C   ADD APRIME SUM AND A SUM TO SUM ARRAY BEFORE INTEGRATING            SUMRMB
C   NOTE: APXYSUM IS 0.0 IF ALPHA1 AND BETA1=0.0                        SUMRMB
      SUM(IS)=SUM(IS) + APXYSUM                                         SUMRMB
C   SUM(IS) AND RSTSUM(IS) MUST BE DIVIDED BY EXP(-SLAML*TAU) IN SUBROU- SUMRMB
C   TINE TRAPINT AND SIMPINT BEFORE THE TOTAL IS INTEGRATED.            SUMRMB
C                                                                       SUMRMB
      RETURN                                                            SUMRMB
      END
```

```
      SUBROUTINE SDBRA(MU,MUP,NU,IC,JC,KC)                              SDBRA
      COMMON/CONFIG/ NP,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)          SDBRA
      COMMON/DEFUNCS/ DPA(448),DMA(448),DBA(448),BPPA(448),BMPA,        SDBRA
     1   BBPA(448),BPMA,BMMA(448),BBMA(448),BPBA(448),BMBA(448),        SDBRA
     2   BBBA(448),INDB,DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,BMB,BBB,   SDBRA
     3   FIMO(14),FIM1MO(14),FJMO(8),FJM1MO(8),FKNO(4),FKM1NO(4),       SDBRA
     4   FKN1(4),FKM1N1(4)                                              SDBRA
      LOGICAL MUSZERO                                                   SDBRA
      DIMENSION B(9)                                                    SDBRA
      EQUIVALENCE(BPP,B(1))                                             SDBRA
C                                                                       SDBRA
C  ONLY COMPUTE THE D AND B FUNCTIONS 1 TIME PER STATE VECTOR,          SDBRA
C  I.E. WHEN IT=1.  THE D AND B ARRAYS WILL CONTAIN THE D AND           SDBRA
C  B FUNCTIONS FOR EACH VECTOR DEFINED BY MU,MUP,NU.  INDB              SDBRA
C  IS THE INDEX INTO THE D AND B ARRAYS -  IT IS ALSO THE               SDBRA
C  COUNTER OF THE MU,MUP,NU VECTORS.                                    SDBRA
C  THE SINGLE VARIABLES DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,           SDBRA
C  BMB,BBB ARE THE WORKING VERSIONS OF THE D AND B ARRAYS, I.E. THE     SDBRA
C  D AND B ARRAYS ARE NEVER MODIFIED DURING I,J,K VECTOR COMPUTA-       SDBRA
C  TIONS.  THEY CHANGE ONLY WHEN (I,J,K) CHANGES.                       SDBRA
C                                                                       SDBRA
C                                                                       SDBRA
C  INITIALIZE D FUNCTIONS TO 0.                                         SDBRA
      DP=0.0                                                            SDBRA
      DM=0.0                                                            SDBRA
      DB=0.0                                                            SDBRA
C  DEFINE COMMON TERMS                                                  SDBRA
      MUS=MU+MUP                                                        SDBRA
      MUSZERO=.FALSE.                                                   SDBRA
      IF(MUS.EQ.0) MUSZERO=.TRUE.                                       SDBRA
C  1/3**(MU+MUP)                                                        SDBRA
      PWRMUS=(1.0/3.0)**MUS                                             SDBRA
      TWODIV3=2.0/3.0                                                   SDBRA
      TMU=2.0*MU                                                        SDBRA
      TMUP=2.0*MUP                                                      SDBRA
      ICM1=IC-1                                                         SDBRA
      JCM1=JC-1                                                         SDBRA
      KCM1=KC-1                                                         SDBRA
      DBCOMI=FIMO(MU+1)                                                 SDBRA
      DBCIM1=FIM1MO(MU+1)                                               SDBRA
      DBCOMJ=FJMO(MUP+1)                                                SDBRA
      DBCJM1=FJM1MO(MUP+1)                                              SDBRA
      DBOK=FKNO(NU+1)                                                   SDBRA
      DBOKM1=FKM1NO(NU+1)                                               SDBRA
      DB1K=FKN1(NU+1)                                                   SDBRA
      DB1KM1=FKM1N1(NU+1)                                               SDBRA
      NPMI=NP-ICM1                                                      SDBRA
      NMMJ=NM-JCM1                                                      SDBRA
      NBMK=NB-KCM1                                                      SDBRA
C                                                                       SDBRA
      MMN=MU+MUP+NU                                                     SDBRA
      IF(MMN.NE.1 .AND. MMN.NE.2) GO TO 50                              SDBRA
C                                                                       SDBRA
C                                                                       SDBRA
C  COMPUTE B FUNCTIONS FOR MU+MUP+NU=1 OR 2                             SDBRA
C  CURRENTLY DEFINITIONS FOR FUNCTIONS BMP AND BPM DO NOT EXIST.        SDBRA
      BMP=0.0                                                           SDBRA
      BPM=0.0                                                           SDBRA
C                                                                       SDBRA
C     DEFINE FUNCTION BPP                                               SDBRA
      BPP=(TMU/NPMI)*DBCIM1*DBCOMJ*DBOK                                 SDBRA
C                                                                       SDBRA
C  DEFINE FUNCTION BMM                                                  SDBRA
      BMM=(TMUP/NMMJ)*DBCOMI*DBCJM1*DBOK                                SDBRA
```

```
C                                                                      SDBRA
C   DEFINE FUNCTION BPB                                                SDBRA
      BBBCOM=(6.0/NBMK)*PWRMUS*DBCOMI*DBCCMJ*DBOKM1                    SDBRA
      BPB=BBBCOM*MU                                                    SDBRA
C                                                                      SDBRA
C   DEFINE FUNCTION BMB                                                SDBRA
      BMB=BBBCOM*MUP                                                   SDBRA
C                                                                      SDBRA
C   DEFINE FUNCTION BBP                                                SDBRA
      BPMCOM=TWODIV3*DB1K*PWRMUS                                       SDBRA
      BBP=BPMCOM*((NPMI-TMU)/NPMI)*DBCIM1*DBCOMJ                       SDBRA
C                                                                      SDBRA
C   DEFINE FUNCTION BBM                                                SDBRA
      BBM=BPMCOM*((NMMJ-TMUP)/NMMJ)*DBCOMI*DBCJM1                      SDBRA
C                                                                      SDBRA
C   DEFINE FUNCTION BBB FOR MU=MUP=0, OTHERWISE BBB=0.0                SDBRA
      BBB=0.0                                                          SDBRA
      IF(MUSZERO) BBB=(2.0/NBMK)*DB1KM1                                SDBRA
C                                                                      SDBRA
      IF(MMN.NE.2) GO TO 150                                           SDBRA
C                                                                      SDBRA
C   DEFINE D FUNCTIONS USING B FUNCTION DEFINITIONS FOR                SDBRA
C   MU+MUP+NU=2                                                        SDBRA
      DP=1.0-(BPP+BMP+BBP)                                             SDBRA
      DM=1.0-(BPM+BMM+BBM)                                             SDBRA
      DB=1.0-(BPB+BMB+BBB)                                             SDBRA
C                                                                      SDBRA
C   ZERO B FUNCTIONS                                                   SDBRA
   50 CONTINUE                                                         SDBRA
      DO 100 IXY=1,9                                                   SDBRA
  100 B(IXY)=0.0                                                       SDBRA
C                                                                      SDBRA
C   PUT D AND B WORKING VARIABLES IN D AND B ARRAYS                    SDBRA
  150 CONTINUE                                                         SDBRA
      DPA(INDB)=DP                                                     SDBRA
      DMA(INDB)=DM                                                     SDBRA
      DBA(INDB)=DB                                                     SDBRA
C                                                                      SDBRA
      BPPA(INDB)=BPP                                                   SDBRA
      BMPA=BMP                                                         SDBRA
      BBPA(INDB)=BBP                                                   SDBRA
      BPMA=BPM                                                         SDBRA
      BMMA(INDB)=BMM                                                   SDBRA
      BBMA(INDB)=BBM                                                   SDBRA
      BPBA(INDB)=BPB                                                   SDBRA
      BMBA(INDB)=BMB                                                   SDBRA
      BBBA(INDB)=BBB                                                   SDBRA
      RETURN                                                           SDBRA
      END                                                              SDBRA
```

```
      SUBROUTINE SDBRB(MU,MUP,NU,IC,JC,KC)                              SDBRB
      COMMON/CONFIG/ NF,NM,NB,NPF,NMF,NBF,NSET(14),QLT(112,51)          SDBRB
      COMMON/DEFUNCS/ DPA(448),DMA(448),DBA(448),BPPA(448),BMPA,        SDBRB
     1   BBPA(448),BPMA,BMMA(448),BBMA(448),BPBA(448),BMBA(448),        SDBRB
     2   BBBA(448),INDB,DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,BMB,BBB,   SDBRB
     3   FIMO(14),FIM1MO(14),FJMO(8),FJM1MO(8),FKNO(4),FKM1NO(4),       SDBRB
     4   FKN1(4),FKM1N1(4)                                              SDBRB
      LOGICAL MUSZERO                                                   SDBRB
      DIMENSION B(9)                                                    SDBRB
      EQUIVALENCE(BPP,B(1))                                             SDBRB
C                                                                       SDBRB
C  ONLY COMPUTE THE D AND B FUNCTIONS 1 TIME PER STATE VECTOR,          SDBRB
C  I.E. WHEN IT=1.  THE D AND B ARRAYS WILL CONTAIN THE D AND           SDBRB
C  B FUNCTIONS FOR EACH VECTOR DEFINED BY MU,MUP,NU.  INDB               SDBRB
C  IS THE INDEX INTO THE D AND B ARRAYS - IT IS ALSO THE                SDBRB
C  COUNTER OF THE MU,MUP,NU VECTORS.                                    SDBRB
C  THE SINGLE VARIABLES DP,DM,DB,BPP,BMP,BBP,BPM,BMM,BBM,BPB,           SDBRB
C  BMB,BBB ARE THE WORKING VERSIONS OF THE D AND B ARRAYS, I.E. THE     SDBRB
C  D AND B ARRAYS ARE NEVER MODIFIED DURING I,J,K VECTOR COMPUTA-       SDBRB
C  TIONS.  THEY CHANGE ONLY WHEN (I,J,K) CHANGES.                       SDBRB
C                                                                       SDBRB
C                                                                       SDBRB
      IF((MU+MUP+NU).EQ.1) GO TO 100                                    SDBRB
      DO 50 IXY=1,9                                                     SDBRB
   50 B(IXY)=0.0                                                        SDBRB
      BPPA(INDB)=BPP                                                    SDBRB
      BMPA=BMP                                                          SDBRB
      BBPA(INDB)=BBP                                                    SDBRB
      BPMA=BPM                                                          SDBRB
      BMMA(INDB)=BMM                                                    SDBRB
      BBMA(INDB)=BBM                                                    SDBRB
      BPBA(INDB)=BPB                                                    SDBRB
      BMBA(INDB)=BMB                                                    SDBRB
      BBBA(INDB)=BBB                                                    SDBRB
C  INITIALIZE D FUNCTIONS TO 0.                                         SDBRB
      DP=0.0                                                            SDBRB
      DM=0.0                                                            SDBRB
      DB=0.0                                                            SDBRB
      IF((MU+MUP+NU).NE.2) GO TO 75                                     SDBRB
      DP=1.0                                                            SDBRB
      DM=1.0                                                            SDBRB
      DB=1.0                                                            SDBRB
   75 DPA(INDB)=DP                                                      SDBRB
      DMA(INDB)=DM                                                      SDBRB
      DBA(INDB)=DB                                                      SDBRB
      RETURN                                                            SDBRB
C  COMPUTE B FUNCTIONS FOR MU+MUP+NU=1.                                 SDBRB
  100 CONTINUE                                                          SDBRB
C  DEFINE COMMON TERMS                                                  SDBRB
      MUS=MU+MUP                                                        SDBRB
      MUSZERO=.FALSE.                                                   SDBRB
      IF(MUS.EQ.0) MUSZERO=.TRUE.                                       SDBRB
C  1/3**(MU+MUP)                                                        SDBRB
      FWRMUS=(1.0/3.0)**MUS                                             SDBRB
      TWODIV3=2.0/3.0                                                   SDBRB
      TMU=2.0*MU                                                        SDBRB
      TMUP=2.0*MUP                                                      SDBRB
      ICM1=IC-1                                                         SDBRB
      JCM1=JC-1                                                         SDBRB
      KCM1=KC-1                                                         SDBRB
```

```
      DBCOMI=FIMO(MU+1)                                                  SDBRB
      DBCIM1=FIM1MO(MU+1)                                                SDBRB
      DBCOMJ=FJMO(MUP+1)                                                 SDBRB
      DBCJM1=FJM1MO(MUP+1)                                               SDBRB
      DBOK=FKNO(NU+1)                                                    SDBRB
      DBOKM1=FKM1NO(NU+1)                                                SDBRB
      DB1K=FKN1(NU+1)                                                    SDBRB
      DB1KM1=FKM1N1(NU+1)                                                SDBRB
      NPMI=NP-ICM1                                                       SDBRB
      NMMJ=NM-JCM1                                                       SDBRB
      NBMK=NB-KCM1                                                       SDBRB
C                                                                        SDBRB
C  DEFINE FUNCTIONS DP,DM AND DB TO BE 0.0                               SDBRB
      DP=0.0                                                             SDBRB
      DPA(INDB)=DP                                                       SDBRB
      DM=0.0                                                             SDBRB
      DMA(INDB)=DM                                                       SDBRB
      DB=0.0                                                             SDBRB
      DBA(INDB)=DB                                                       SDBRB
C                                                                        SDBRB
C                                                                        SDBRB
C  CURRENTLY DEFINITIONS FOR FUNCTIONS BMP AND BPM DO NOT EXIST.         SDBRB
      BMP=0.0                                                            SDBRB
      BMPA=BMP                                                           SDBRB
      BPM=0.0                                                            SDBRB
      BPMA=BPM                                                           SDBRB
C                                                                        SDBRB
C     DEFINE FUNCTION BPP                                                SDBRB
      BPP=(TMU/NPMI)*DBCIM1*DBCOMJ*DBOK                                  SDBRB
      BPPA(INDB)=BPP                                                     SDBRB
C                                                                        SDBRB
C  DEFINE FUNCTION BMM                                                   SDBRB
      BMM=(TMUP/NMMJ)*DBCOMI*DBCJM1*DBOK                                 SDBRB
      BMMA(INDB)=BMM                                                     SDBRB
C                                                                        SDBRB
C  DEFINE FUNCTION BPB                                                   SDBRB
      BBBCOM=(6.0/NBMK)*PWRMUS*DBCOMI*DBCOMJ*DBOKM1                      SDBRB
      BPB=BBBCOM*MU                                                      SDBRB
      BPBA(INDB)=BPB                                                     SDBRB
C                                                                        SDBRB
C  DEFINE FUNCTION BMB                                                   SDBRB
      BMB=BBBCOM*MUP                                                     SDBRB
      BMBA(INDB)=BMB                                                     SDBRB
C                                                                        SDBRB
C  DEFINE FUNCTION BBP                                                   SDBRB
      BPMCOM=TWODIV3*DB1K*PWRMUS                                         SDBRB
      BBP=BPMCOM*((NPMI-TMU)/NPMI)*DBCIM1*DBCOMJ                         SDBRB
      BBPA(INDB)=BBP                                                     SDBRB
C                                                                        SDBRB
C  DEFINE FUNCTION BBM                                                   SDBRB
      BBM=BPMCOM*((NMMJ-TMUP)/NMMJ)*DBCOMI*DBCJM1                        SDBRB
      BBMA(INDB)=BBM                                                     SDBRB
C                                                                        SDBRB
C  DEFINE FUNCTION BBB FOR MU=MUP=0, OTHERWISE BBB=0.0                   SDBRB
      BBB=0.0                                                            SDBRB
      IF(MUSZERO) BBB=(2.0/NBMK)*DB1KM1                                  SDBRB
      BBBA(INDB)=BBB                                                     SDBRB
      RETURN                                                             SDBRB
      END                                                                SDBRB
```

```
      FUNCTION PCOND(MU,MUP,NU,IC,JC,KC,IT)                              PCOND
      COMMON/INVAR/ EMLAM(3,51),EMDEL(4,51),EMLAM1(3,51),EMLAM2(3,51),   PCOND
     1              G2(3,51),AT(3,51),CT(3,51)                           PCOND
      REAL I2CPSQ,J2CMSQ,K2CBSQ                                          PCOND
C                                                                        PCOND
C   AT TIME 0 PCOND=1.0 FOR MU=IC,MUP=JC AND NU=KC; PCOND=0.0 OTHERWISE. PCOND
      IF(IT.GT.1) GO TO 10                                               PCOND
      IF(MU.EQ.IC .AND. MUP.EQ.JC .AND. NU.EQ.KC) GO TO 20               PCOND
      PCOND=0.0                                                          PCOND
      RETURN                                                             PCOND
   20 PCOND=1.0                                                          PCOND
      RETURN                                                             PCOND
C                                                                        PCOND
   10 CONTINUE                                                           PCOND
C   PCOND=0 IF MU.GT.IC OR MUP.GT.JC OR NU.GT.KC                         PCOND
      IF(MU.LE.IC .AND. MUP.LE.JC .AND. NU.LE.KC) GO TO 30               PCOND
      PCOND=0.0                                                          PCOND
      RETURN                                                             PCOND
C                                                                        PCOND
   30 CONTINUE                                                           PCOND
      IP=1                                                               PCOND
      IM=2                                                               PCOND
      IB=3                                                               PCOND
      CP=CT(IP,IT)                                                       PCOND
      CPSQR=CP*CP                                                        PCOND
      CM=CT(IM,IT)                                                       PCOND
      CMSQR=CM*CM                                                        PCOND
      CB=CT(IB,IT)                                                       PCOND
      CBSQR=CB*CB                                                        PCOND
C                                                                        PCOND
C   COMPUTE BINOMIAL COEFFICIENTS IN THE NUMERATOR                       PCOND
      BINOMC=FNCK(IC,MU)*FNCK(JC,MUP)*FNCK(KC,NU)                        PCOND
C                                                                        PCOND
      XNUMER=BINOMC*(CP**MU) * (CM**MUP) * (CB**NU)                      PCOND
      I2CPSQ=0.0                                                         PCOND
      IF(IC.GE.2) I2CPSQ=FNCK(IC,2)*CPSQR                                PCOND
      J2CMSQ=0.0                                                         PCOND
      IF(JC.GE.2) J2CMSQ=FNCK(JC,2)*CMSQR                                PCOND
      K2CBSQ=0.0                                                         PCOND
      IF(KC.GE.2) K2CBSQ=FNCK(KC,2)*CBSQR                                PCOND
C                                                                        PCOND
      DENOMR=1.0+(IC*CP)+I2CPSQ+(JC*CM)+J2CMSQ+(KC*CB)+K2CBSQ+           PCOND
     1          (IC*JC*CP*CM)+(IC*KC*CP*CB)+(JC*KC*CM*CB)                PCOND
      PCOND=XNUMER/DENOMR                                                PCOND
      RETURN                                                             PCOND
      END                                                                PCOND
```

```
      SUBROUTINE EIGEN(DELTA1,DELTA2)                                   EIGEN
C     ***********************************************                   EIGEN
C       EIGEN CALLS TWO IMSL SUBRCUTINES--EIGRF & LEQT2F                EIGEN
C       EIGRF COMPUTES EIGENVALES AND EIGENVECTORS OF                   EIGEN
C       THE MATRIX EIGA(I,J)                                            EIGEN
C       LEQT2F SOLVES THE LINEAR SYSTEM AX=Y, WHERE                     EIGEN
C       THE COLUMNS OF A ARE THE EIGENVECTORS OF EIGA                   EIGEN
C       AND Y IS AN M BY N MATRIX WHOSE COLUMNS ARE THE INDIVI-         EIGEN
C       DUAL RIGHT HAND SIDES(NON-HOMOGENCUS TERMS)                     EIGEN
C     ***********************************************                   EIGEN
      COMMON/RATES/ LAMP,LAMM,LAMB,LAMBG,DELTAP,DELTAM,DELTAB,DELTABG,  EIGEN
     1              ALPHA1,BETA1,ALPHA2,BETA2                           EIGEN
      COMMON/EIGCOM/ EIGSD(3,3,3),EIGWR(3),G2WT(9,51),H2WT(9,51),       EIGEN
     1              G2PWT(9,51),H2PWT(9,51)                             EIGEN
      DIMENSION WK(15),WKAREA(18),A(3,3),EIGC(3,3),EIGA(3,3)            EIGEN
      REAL LAMP,LAMM,LAMB,LAMBG                                         EIGEN
      COMPLEX W(3),Z(3,3)                                               EIGEN
C       NEIG IS THE SIZE OF THE MATRIX WHOSE EIGENVALUES                EIGEN
C       WE ARE FINDING                                                  EIGEN
C       15(WK DIMENSION) IS OBTAINED BY MULT. NEIG+2 BY NEIG            EIGEN
C       18(WKAREA DIMENSION) IS OBTAINED BY EVALUATING (NEIG**2)+3*NEIG EIGEN
      NEIG=3                                                            EIGEN
      M=3                                                               EIGEN
      IDGT=4                                                            EIGEN
      IJOB=1                                                            EIGEN
C       EIGA(I,J) ARE THE ELEMENTS OF THE MATRIX WHOSE                  EIGEN
C       EIGENVALUES WE ARE FINDING                                      EIGEN
      EIGA(1,1)=-(ALPHA1+DELTA1+BETA2)                                  EIGEN
      EIGA(2,1)=ALPHA1                                                  EIGEN
      EIGA(3,1)=0.                                                      EIGEN
      EIGA(1,2)=BETA1                                                   EIGEN
      EIGA(2,2)=-(BETA1+BETA2)                                          EIGEN
      EIGA(3,2)=BETA2                                                   EIGEN
      EIGA(1,3)=0.                                                      EIGEN
      EIGA(2,3)=ALPHA2                                                  EIGEN
      EIGA(3,3)=-(ALPHA2+DELTA2+BETA1)                                  EIGEN
C       EIGC(I,J) IS THE MATRIX WHOSE COLUMNS ARE THE VECTORS           EIGEN
C      ON THE RIGHT HAND SIDE(NON-HOMOGENEOUS TERMS) OF THE             EIGEN
C      MATRIX EQUATION TO BE SOLVED BY LEQT2F                           EIGEN
      DO 50 J=1,3                                                       EIGEN
      DO 50 I=1,3                                                       EIGEN
      EIGC(I,J)=0.                                                      EIGEN
   50 IF(I.EQ.J) EIGC(I,J)=1.                                           EIGEN
      CALL EIGRF(EIGA,NEIG,NEIG,IJOB,W,Z,NEIG,WK,IER)                   EIGEN
      DO 60 I=1,NEIG                                                    EIGEN
      EIGWR(I)=W(I)                                                     EIGEN
      DO 70 J=1,NEIG                                                    EIGEN
      EIGA(I,J)=Z(I,J)                                                  EIGEN
      IF(ABS(EIGA(I,J)).GT.1.0E-10) GO TO 65                            EIGEN
      EIGA(I,J)=0.0                                                     EIGEN
 65   A(I,J)=EIGA(I,J)                                                  EIGEN
 70   CONTINUE                                                          EIGEN
 60   CONTINUE                                                          EIGEN
      CALL LEQT2F(A,M,NEIG,NEIG,EIGC,IDGT,WKAREA,IER)                   EIGEN
      DO 80 K=1,NEIG                                                    EIGEN
      DO 80 I=1,NEIG                                                    EIGEN
      DO 80 J=1,NEIG                                                    EIGEN
C  EIGSD(I,J,K) ARE THE CONSTANTS USED TO CONSTRUCT THE                 EIGEN
C  PROBABILITY FUNCTIONS P(I/K(T)) - PROBABILITY THAT THE SYSTEM        EIGEN
C  IS IN STATE I AFTER STARTING IN STATE K.                             EIGEN
      EIGSD(I,J,K)=EIGA(I,J)*EIGC(J,K)                                  EIGEN
C                                                                       EIGEN
 80   CONTINUE                                                          EIGEN
      RETURN                                                            EIGEN
      END                                                               EIGEN
```

REFERENCES

1. Henrici, P., Elements of Numerical Analysis, John Wiley & Sons, Inc., New York, 1964.

2. Isaacson, Eugene and Herbert B. Keller, Analysis of Numerical Methods, John Wiley & Sons, Inc., New York, 1966.

3. Ostrowski, A. M., Solutions of Equations and Systems of Equations, Academic Press, New York, 1966.

4. Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1963.

5. ---------, "Error Analysis of Eigenvalue Techniques Based On Orthogonal Transformations," J. Soe. Induc. Appl. Math. Vol. 10, pp. 162-195 (1962).

6. ---------, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

7. Papathomas, T. and O. Wing, "Sparce Hessenberg Reduction and the Eigenvalue Problem for Large Sparce Matrices," IEEE Transactions on Circuits and Systems, Vol. Cas-23, No. 12, pp. 739-744 (1976).

8. ---------, "A Scheme for the Solution of the Eigenvalue Problem for Large Sparce Matrices," Proceedings of the 1976 IEEE International Symposium on Circuits and Systems, Munich, Germany, 27-29 April, 1976 (New York, USA: IEEE 1976), pp. 259-262.

9. ---------, "Hessenberg Reduction Methods for Sparce Circuits and Systems," 1977 IEEE International Symposium on Circuits and Systems Proceedings, Phoenix, Arizona, USA, 25-27 April 1977 (New York, USA: IEEE 1977), pp. 85-88.

10. Parlett, "Laguerre's Method Applied to the Matrix Eigenvalue Problem," Math. Comp., Vol. 18, pp. 464-485 (1964).

REFERENCES (Cont.)

11. Parlett, B. and C. Reinsch, "Balancing a Matrix For Calculation of Eigenvalues and Eigenvectors," Nurm. Math., Vol. 13, pp. 293-304 (1969).

12. Tewarson, R. P., "On the Reduction of a Sparce Matrix to Hessenberg Form," Int. J. of Computer Math., Vol. 2 pp. 283-295 (1970).